

UNIVERSIDAD CARLOS III DE MADRID
ESCUELA POLITÉCNICA SUPERIOR
GRADO EN INGENIERÍA INFORMÁTICA

DESARROLLO DE UN ENTORNO PARA EL DISEÑO COLABORATIVO

Sara Lucía Pérez García

Tutor: Rosa María Romero Gómez

Director: David Díez Cebollero

09 de julio de 2013

Agradecimientos

A Fer, por su apoyo constante en los últimos años, y por todos sus “venga, que ya no queda nada”.

A Je, por encontrarnos en el baño el primer día de universidad y compartirlo todo desde entonces.

A Lydi, por ser la mejor hermana pequeña del mundo.

A mi familia y amigos, porque sé lo orgullosos que se sienten de que haya llegado hasta aquí.

Resumen

El presente trabajo fin de grado se desarrolla dentro del contexto del co-diseño. El concepto de co-diseño puede definirse como *“un proceso de colaboración creativa entre diseñadores y personas, los cuáles no poseen una formación previa en diseño, con el propósito de resolver problemas significativos”*. Dentro de este contexto, en los últimos tiempos, ha surgido una tendencia de desarrollo de herramientas tecnológicas que mejoren dicho proceso de diseño. Sin embargo, el soporte a los procesos de co-diseño llevados a cabo de manera distribuida y/o síncrona supone un desafío a encarar.

En ocasiones, es necesario que este proceso de co-diseño pueda ser realizado de manera distribuida, con el fin de facilitar la participación de las personas desde diferentes lugares. De manera similar, existe la necesidad de obtener recursos de diseño de manera remota. En particular, los participantes en el proceso de co-diseño pueden querer obtener recursos multimedia aunque estos no se encuentren en el mismo espacio de diseño. De acuerdo a esto, es necesario asistir al proceso de co-diseño de manera distribuida y asegurar que, en ambos casos, existe una coordinación durante el acceso a los datos.

A fin de solventar dicha problemática, el presente trabajo fin de grado provee un entorno tecnológico que soporta el proceso de co-diseño. Concretamente, las contribuciones de este proyecto versan tanto sobre la asistencia a la creación de diagramas de afinidad de manera distribuida, como sobre la recogida de información multimedia de forma remota. Con este propósito, en este proyecto se extiende la funcionalidad de la herramienta de diseño ColDes [1], permitiendo al usuario crear un nuevo tipo de sala de diseño que soporta la creación de diagramas de afinidad de manera distribuida. Ahora, los participantes pueden crear y compartir ideas de diseño añadiendo ítems de información relacionados con el problema establecido en el proceso de diseño. Estos ítems de información pueden ser organizados en grupos de ideas similares que sean significativos para la resolución del problema. De esta manera, se obtiene una representación visual del primer esbozo de la solución de diseño. Con el objetivo de permitir la obtención de recursos multimedia de manera remota, se ha desarrollado una aplicación móvil. Los participantes pueden tomar fotografías y vídeos desde cualquier lugar y añadirlos a los diagramas de afinidad. Estos recursos se pueden utilizar para afianzar las ideas plasmadas en el diagrama de afinidad, ya que es posible asociarlos a los ítems de información creados en los *workspace* de diagrama de afinidad de ColDes.

Toda la información que se maneja en el sistema de ColDes es persistente. El usuario puede guardar en todo momento el estado de un diagrama de afinidad y cargarlo posteriormente para continuar con el trabajo. El usuario, por tanto, puede realizar su trabajo en los diagramas de afinidad de manera discontinúa. Dado que varios usuarios pueden acceder al mismo tiempo a los

recursos del sistema, existen políticas de colaboración que aseguran la consistencia de los datos en los casos en los que el proceso se realiza de manera distribuida y/o síncrona.

Abstract

This final project is framed within the context of co-design. The concept of co-design can be defined as *“a process of creative collaboration between designers and people, which have no previous training in design, in order to solve meaningful problems”*. Within this context, there exists a recent trend of development of technological tools that improve the process of design. However, the support of the co-design processes carried out in a distributed and/or synchronous manner presents a challenge to be solved.

It is often necessary that this process of co-design can be performed in a distributed manner in order to facilitate the participation of people from different locations. Similarly, there is a need to obtain design resources remotely. In particular, the participants may want to get multimedia resources even if these are not in the same design space. Accordingly, it is necessary to assist the co-design process in a distributed manner and to ensure that, in both cases, there is a need of coordination for the data access.

In order to address these requirements, this final project provides a technological environment that supports the co-design process. In particular, the contributions of this project are related to both the assistance of building affinity diagrams in a distributed manner and collecting multimedia information remotely. With this purpose, this project extends the functionality of a design tool called ColDes [1] by allowing to create a new type of design room that support the creation of affinity diagrams in a distributed manner. The participants can now create and share design ideas by adding information item related to the problem established in the design process. These information items can be organized into groups of similar ideas that are meaningful to the resolution of the problem. In this way, it is obtained a visual representation of the first draft of the design solution. Aiming at allowing obtaining multimedia resources remotely, it has been developed a mobile application. The participants can take photos and videos from anywhere and add them to the affinity diagram. These resources can be used to reinforce ideas of the affinity diagram, since it is possible to associate them to the information items created in the affinity diagram workspace of ColDes.

To conclude, all the information stored in the ColDes system has been made persistent. The participants can save an affinity diagram at any time, and recover it later to continue the design. They, therefore, can perform their work in an affinity diagram in a discontinued manner. Since several users can simultaneously access the system resources, some collaboration policies have been introduced to ensure the consistency of the data in cases where the process is performed in a distributed and/or synchronous manner.

Índice de contenidos

Agradecimientos	2
Resumen	3
Abstract	5
Índice de contenidos	6
Índice de figuras.....	8
Índice de tablas.....	10
Glosario de términos y acrónimos.....	11
1 Introduction	12
1.1 Problem definition	12
1.2 Aims	13
1.3 Development phases.....	14
1.4 Project resources	14
1.5 Document structure	15
2 El estado de la cuestión.....	16
2.1 Revisión teórica de técnicas de co-diseño.....	16
Elección de una técnica de co-diseño.....	26
2.2 Revisión de tecnologías móviles	27
Criterios de evaluación	28
Evaluación cualitativa	29
Evaluación cuantitativa.....	32
3 Gestión de proyecto software	34
3.1 Estimación de recursos	34
Costes de personal	34
Costes de hardware y software	37
Gastos generales	38
3.2 Presupuesto	39
4 Solución	41
4.1 Descripción de la solución.....	41

Aplicación web.....	41
Aplicación móvil.....	43
4.2 El proceso de desarrollo.....	44
Modelo de proceso.....	45
Análisis.....	46
Definición de requisitos.....	46
Casos de uso.....	51
Diseño.....	61
Diseño de sistema	61
Diseño detallado	65
Implementación.....	83
Entorno de desarrollo	83
Organización del proyecto	85
5 Evaluación.....	95
5.1 Proceso de evaluación	95
Plan de pruebas	95
Casos de prueba	96
5.2 Análisis de resultados	117
6 Conclusions	119
6.1 Contributions	119
6.2 Future work.....	119
6.3 Problems encountered.....	120
6.4 Personal opinions.....	120
7 Bibliografía.....	122
Anexo I. Control de versiones	125
Anexo II. Seguimiento del trabajo fin de grado	127
Forma de seguimiento.....	127
Planificación del trabajo fin de grado	127
Anexo III. Especificación de requisitos.....	129
Requisitos funcionales.....	129
Requisitos no funcionales	136
Anexo IV. Prototipo	139
Aplicación móvil	139

Índice de figuras

Figura 1: <i>Time/space groupware matrix</i> .	13
Figura 2: Grupo de trabajo realizando <i>group sketching</i> .	18
Figura 3: Ejemplos de <i>rough prototyping</i> .	19
Figura 4: <i>Affinity diagram</i> .	21
Figura 5: Referencias cruzadas en una <i>motivation matrix</i> .	22
Figura 6: <i>Mind map</i> .	23
Figura 7: <i>Storyboard</i> de una aplicación móvil.	25
Figura 8: Técnicas de co-diseño según los requisitos del proyecto.	27
Figura 9: Aplicación web. <i>Workspace</i> para la creación de diagramas de afinidad.	43
Figura 10: Aplicación móvil. Administración de recursos.	44
Figura 11: Esquema del funcionamiento de un ciclo iterativo incremental.	45
Figura 12: Distribución de la cuota de mercado entre las diferentes versiones de Android.	62
Figura 13: Arquitectura del entorno desarrollado.	63
Figura 14: Diagrama de secuencia: administración de recursos.	65
Figura 15: Estructura de directorios del sistema de ficheros.	66
Figura 16: Diagrama físico de datos.	67
Figura 17: Diagrama de clases: Utilización de patrones de diseño.	75
Figura 18: Prototipo del <i>workspace</i> de diagrama de afinidad. Primera iteración (izquierda).	77
Figura 19: Interfaz del <i>workspace</i> de diagrama de afinidad. Aplicación final.	78
Figura 20: Prototipo de la pantalla principal. Primera iteración (izquierda) y segunda iteración (derecha).	79
Figura 21: Prototipo de la interfaz de subida de recursos. Primera iteración (izquierda) y segunda iteración (derecha).	80
Figura 22: Prototipo de la interfaz de gestión de etiquetas. Segunda iteración.	80
Figura 23: Interfaz de subida de recursos. Aplicación final.	81
Figura 24: Prototipo de la interfaz del módulo de etiquetas. Segunda iteración (izquierda) y aplicación final (derecha).	82
Figura 25: Prototipo de la interfaz del módulo de <i>workspace</i> . Segunda iteración.	83
Figura 26: IDE Eclipse con ADT <i>plug-in</i> .	84
Figura 27: IDE NetBeans.	85
Figura 28: Organización general del proyecto para la aplicación web.	86
Figura 29: Organización del proyecto para la aplicación web. Parte servidora.	87
Figura 30: Organización del proyecto para la aplicación web. Parte cliente.	88
Figura 31: Organización general del proyecto para la aplicación móvil.	89
Figura 32: Organización del proyecto para la aplicación móvil. Directorio <i>src</i> .	90
Figura 33: Organización del proyecto para la aplicación móvil. Directorio <i>res</i> .	92
Figura 34: Organización del proyecto para el <i>Web service</i> .	93
Figura 35: Organización de la librería <i>ColDesDAO</i> .	94
Figura 36: Diagrama de Gantt: Planificación inicial del trabajo fin de grado.	127
Figura 37: Prototipo de la interfaz del módulo de autenticación de la aplicación móvil. Segunda iteración.	140
Figura 38: Prototipo de la interfaz del módulo de recursos de la aplicación móvil. Segunda iteración.	141
Figura 39: Prototipo de la interfaz de la vista de recursos de la aplicación móvil. Segunda iteración.	142

Figura 40: Prototipo de la interfaz del módulo de etiquetas de la aplicación móvil. Segunda iteración.	142
Figura 41: Prototipo de la interfaz del módulo de usuario de la aplicación móvil. Segunda iteración.	143
Figura 42: Prototipo de la interfaz del módulo de <i>workspaces</i> de la aplicación móvil. Segunda iteración. ..	143
Figura 43: Prototipo de la interfaz de la vista de <i>workspace</i> de la aplicación móvil. Segunda iteración.	144

Índice de tablas

Tabla 1: Resumen del análisis teórico de las tecnologías.	29
Tabla 2: Comparación ponderada de las tecnologías.	32
Tabla 3: Salario bruto mensual del equipo de trabajo.	36
Tabla 4: Coste total de personal.	36
Tabla 5: Coste imputable de recursos hardware y software.	38
Tabla 6: Coste de infraestructuras y comunicaciones.	38
Tabla 7: Coste total asociado al proyecto.	39
Tabla 8: Coste final asociado al proyecto.	39
Tabla 9: Plazos de pago del coste final del proyecto.	39
Tabla 10: Plantilla de especificación de requisitos.	47
Tabla 11: Plantilla de especificación de casos de uso.	51
Tabla 12: Matriz de trazabilidad: Requisitos funcionales vs. Casos de uso. Aplicación web.	60
Tabla 13: Matriz de trazabilidad: Requisitos funcionales vs. Casos de uso. Aplicación móvil.	61
Tabla 14: Descripción de la tabla <i>userresource</i> de la base de datos.	68
Tabla 15: Descripción de la tabla <i>roomresource</i> de la base de datos.	69
Tabla 16: Descripción de la tabla <i>usertag</i> de la base de datos.	69
Tabla 17: Descripción de la tabla <i>tagresource</i> de la base de datos.	70
Tabla 18: Descripción de la tabla <i>roomitem</i> de la base de datos.	70
Tabla 19: Descripción de la tabla <i>itemgroup</i> de la base de datos.	71
Tabla 20: Descripción de la tabla <i>groupitem</i> de la base de datos.	72
Tabla 21: Descripción de la tabla <i>itemresource</i> de la base de datos.	72
Tabla 22: Plantilla de especificación de casos de prueba.	95
Tabla 23: Resultados del proceso de evaluación.	118
Tabla 24: Control de versiones.	126

Glosario de términos y acrónimos

Catálogo de términos y acrónimos específicos del contexto del trabajo.

API:	<i>Application Programming Interface</i> (Interfaz de Programación de Aplicaciones). Conjunto de funciones y procedimientos que ofrece una biblioteca para ser utilizado por otro software en una capa superior.
CSS:	<i>Cascading Style Sheets</i> (Hojas de Estilo en Cascada). Lenguaje utilizado para definir el estilo de un documento HTML.
Framework:	Estructura conceptual y tecnológica de soporte definido, con el fin de que un proyecto software pueda ser organizado y desarrollado.
IDE:	<i>Integrated Development Environment</i> (Entorno de Desarrollo Integrado). Software compuesto de un conjunto de herramientas de programación que permite al programador crear aplicaciones a partir de un lenguaje de programación.
MySQL:	Sistema de gestión de base de datos relacional, multihilo y multiusuario.
Plug-in:	Es un complemento que se relaciona con una aplicación para aportarle una nueva funcionalidad
REST:	<i>Representational State Transfer</i> es una técnica utilizada para la comunicación entre sistemas hipermedia distribuidos. Se utiliza como protocolo para el diseño de servicios web (<i>Web services</i>).
SOAP:	<i>Simple Object Access Protocol</i> es un protocolo utilizado para el diseño de <i>Web services</i> , y que permite el intercambio de datos entre aplicaciones a partir de ficheros con estructura XML.
SDK:	<i>Software Development kit</i> (Kit de Desarrollo de Software). Conjunto de herramientas de desarrollo de software que permiten al programador crear aplicaciones para un sistema concreto.
XML:	Lenguaje de programación por etiquetas desarrollado como estándar para el intercambio de información estructurada.
Web service:	Tecnología que permite el intercambio de datos entre aplicaciones a partir de un conjunto de protocolos y estándares.

1 Introduction

Co-design is referred as *“The creativity of designers and people not trained in design working together in the design development process.”* [2, p. 6]. The co-design concept can be understood, therefore, as a creative activity in which groups of people, who do not have to have a background in design, come together to solve meaningful problems. Accordingly, in the context of this work, design process is used to refer to the creativity of designers and people not trained in design working together in the design development process.

In recent years, it has emerged a tendency of developing technologies and tools aimed at enhancing this design process, which imposes a technological challenge. Along this document, it has defined the process of creating a technology tool that supports the co-design process.

1.1 Problem definition

In order to support the comprehension of the problem to be addressed by this work, it is necessary to review some previous concepts related to the co-design process. At first, it is necessary to define the concept of collaboration as *“the act of working with another or others on a joint project”* [3]. This concept is highly related to the concept of *groupware*, which is defined as a *“software systems specifically designed to support group working with cooperative requirements in mind”* [4]. Particularly, *groupware* systems can be classified working on different space and time circumstances. The space dimension is divided into co-located (same place) and remote (different place). The time dimension is divided into synchronous (same time) and asynchronous (different time) systems. [Figura 1](#) shows the time/space groupware matrix.

This work is particularly focused on the development of systems that allow work synchronously and distributed. In what follows, different technological challenges imposed by these systems are described.

Firstly, it's necessary to ensure data integrity when several participants need to access to data at the same time and can making changes to them. Similarly, when multiple participants needs to access to this data from different locations, it is necessary to preserve the coordination between them. Therefore, the support of this co-design process in a distributed way constitutes the main problem to deal with in the context of this work. In particular, and according to the strictest conception of the co-design process, different people working from different locations are not allowed to using the same resource for co-design. Therefore, it's necessary to solve the problems outlined above in order to allow people work synchronously and in distributed manner.

Secondly, it's necessary to consider the contextual environment of the process of co-design. Participants sometimes need to use external resources into the process of co-design. These resources are available from different places and from the place where the process of co-design

occurs. Accordingly, the above problems must be solved to enable different participants to use resources obtained from different places.

	Same time	Different time
Same place	Dialogues Informal meetings Formal team meetings	File management Bulletin board Paper project dossier
Different place	Telephone Tele conferencing Video conferencing Instant messaging	Postal + Interoffice mail Facsimile Computer network Email MS-outlook calendar

Figura 1: *Time/space groupware matrix*¹.

1.2 Aims

The purpose of this work is to provide a technological environment that supports the co-design process. This aim can be depicted in two specific goals:

- Building affinity diagrams in a distributed manner.
- Collecting multimedia information in a contextual manner.

Aiming at meeting the first goal, a tool called ColDes [1] has been used as a basis. This tool, previously developed in a final project degree, now allows to create a new type of design room that support the creation of affinity diagrams in a distributed manner.

With the purpose of meeting the second goal, a mobile application has been developed in order to allow participants to obtain real-time multimedia resources. These multimedia resources can be used in the co-design activities assisted by ColDes. Thus, it's not necessary to have all the design elements in the place which is the design space.

¹ Fuente: <http://www.emeraldinsight.com/journals.htm?articleid=1626459&show=html>

1.3 Development phases

This section defines the different phases of the development process followed along the project. In particular, these phases are: (1) the study of the problem; (2) the definition of the problem; (3) the development of the solution; and (4) the evaluation of the solution.

The first phase, the study of the problem, seeks to explore the context in which the project is developed. In particular, this phase establishes the deficiencies found within this context. These deficiencies are thus that the project should address. In this phase, the state of the art of different co-design techniques and mobile technologies is reviewed. Based on this review, the most suitable techniques and technologies are identified.

The second phase, the definition of the problem, seeks to expose deficiencies found in the previous phase. These deficiencies are presented as the problem that the project must address.

The third phase, the development of the solution phase, presents the solution of this project, which seeks to address the deficiencies previously mentioned. In the context of this project, there is a first stage of analysis, which describes the functionality that the solution needs to satisfy. Secondly, there is a design stage, which defines models or representations of the entities that will be included in the system. In the third stage, the implementation process is described.

To conclude, the evaluation of the solution aims to demonstrate the validity of the developed solution. To validate the solution, it is necessary to address the requirements defined in the previous phase.

1.4 Project resources

This section lists the resources used in the development of the project. These resources are categorized under two categories: technological or conceptual.

Technological resources refers to the hardware and software tools used for writing this document and developing the solution. Specifically, it has been used a personal computer and a smartphone for the execution of the mobile application.

Conceptual resources refers to previous works that has served as a basis for the development of this project. In this case, it has been used the bibliographic catalog of the library of the Universidad Carlos III de Madrid (UC3M) and academic databases such as Google Scholar.

1.5 Document structure

This section defines the structure of this document.

The first chapter serves as an introductory chapter, presenting the definition of the problem to be addressed and the goals to be achieved.

In the second chapter, the state of the art of different co-design techniques and mobile technologies is reviewed.

Then, the third chapter describes the management of this project. In particular, this chapter establishes the project planning and estimates the needed resources.

Following, the fourth chapter presents the developed solution. In particular, this chapter describes the different stages of the development process: analysis, design implementation, and evaluation.

Next, technical conclusions and personal opinions are drawn in the fifth chapter. This chapter also establishes the main contributions of the project, and the future works that emerge from this project.

2 El estado de la cuestión

En el estado de la cuestión se estudia el contexto de trabajo del proyecto. De acuerdo a esto, el presente capítulo revisa diversas técnicas de soporte al concepto de co-diseño. A continuación, se escoge la más adecuada para completar la aplicación ColDes creando un espacio de trabajo que asista el co-diseño. Asimismo, se realiza una revisión de tecnologías móviles a fin de determinar la más adecuada para desarrollar una aplicación móvil que permita recopilar información multimedia de manera contextual.

En el artículo *Co-creation and the new landscapes of design* se define co-diseño como sigue: “We use co-design in a broader sense to refer to the creativity of designers and people not trained in design working together in the design development process.” [2, p. 6] Se entiende, por tanto, el proceso de diseño fuera de su definición clásica para introducir la participación de personas no instruidas en diseño en el proceso de desarrollo del diseño.

En la actualidad, se puede definir la innovación en términos de la demanda, en lugar de en términos de la oferta. Esto quiere decir que el valor y la satisfacción obtenidos de recursos del propio cliente está muy vigente. En este contexto, los procesos de diseño centrados en el usuario ofrecen grandes ventajas con respecto a los sistemas de diseño centrados en el fabricante. En este contexto, el co-diseño implica varios factores. En primer lugar, supone escuchar las opiniones del cliente, analizarlas y tomarlas en cuenta. Por otra parte, supone abrir la mentalidad para comprender sus ideas y llevarlas a cabo. En definitiva, implica que el cliente debe colaborar en el proceso de desarrollo del producto o, en este caso, del sistema, entendiendo colaboración como la suma de individuos que se unen para llegar a un fin.

2.1 Revisión teórica de técnicas de co-diseño

Una técnica de co-diseño es aquel procedimiento que permite llevar a la práctica el proceso de co-diseño. Es decir, permite asistir la colaboración entre desarrolladores y personas no instruidas en diseño en el proceso de desarrollo del diseño [5].

En la presente sección se realiza una revisión de las técnicas de co-diseño más relevantes: **Group sketching**, **rough prototyping**, **affinity diagram**, **motivation matrix**, **mind map**, **storyboard** y **story telling**. A continuación, se presenta una descripción para cada una de las herramientas, acompañada de una imagen que ayude a ilustrar la definición.

Group sketching

Esta técnica se basa en la creación de dibujos básicos y sencillos, con el fin de fomentar la participación de todos los miembros del equipo de trabajo. Es una técnica rápida, fácil y económica para el desarrollo y la explicación de ideas por varios participantes de manera simultánea. Se utiliza en el proceso de co-diseño para compartir los conocimientos dentro del

equipo. Ofrece una base común para fomentar la discusión, incluso cuando los participantes tienen antecedentes culturales y sociales diferentes. En concreto, se facilita la colaboración a través de las siguientes características:

- Permite a todos los participantes compartir una visión común de la superficie de trabajo mientras que proporciona acceso simultáneo a la misma.
- Entremezcla las acciones realizadas sobre la superficie de trabajo (listas, dibujos, gestos) y las funciones que representan (almacenamiento de la información, expresión de ideas).
- Proporciona medios de apoyo a la comunicación gestual. Los gestos deben ser claramente visibles, y deben mantener relación con los objetos dentro de la superficie de trabajo y con la comunicación mediante la voz.
- Permitir que el proceso de expresión de ideas se realice mediante la transmisión de pequeños cambios en la actividad del usuario.

Casi todos los procesos de grupo comienzan con una serie de reuniones de diseño iniciales, donde los participantes pueden expresarse, discutir y desarrollar sus ideas. Los participantes suelen utilizar algún tipo de superficie de trabajo común para facilitar la interacción, como pizarras, u hojas grandes de papel para dibujar y tomar notas. Dado que el trabajo de un individuo está cada vez más centrado en torno a un ordenador en red, éste puede llegar a ser un medio valioso para que personas situadas en diferentes ubicaciones puedan compartir su trabajo en línea a través de una superficie de trabajo electrónico.

Teniendo en cuenta estas características, surgen herramientas como *GroupSketch* [6], que se puede entender como un bloc de dibujo multiusuario que soporta las actividades de diseño en tiempo real por grupos pequeños.



Figura 2: Grupo de trabajo realizando *group sketching*².

Rough prototyping

Esta técnica permite crear prototipos de manera rápida y sencilla utilizando cualquier tipo de objetos y materiales disponibles en el espacio y lugar en el cual se realiza la actividad de diseño. Concretamente, esta técnica asiste a la visualización de ideas y supone una forma de asegurarse de que todos los miembros del equipo traten el problema desde el mismo punto de vista. Asimismo, contribuye a que el proceso de diseño sea más interactivo y concreto.

Los elementos utilizados simulan los componentes que forman parte del contexto del problema, con el fin de expresar mejor una idea frente al resto de miembros del equipo. En el diseño de software, los prototipos se suelen realizar en papel, aunque resulta muy útil tener acceso a otros objetos y materiales, ya que supone una forma de ayudar a estimular nuevas ideas. Esta técnica es especialmente útil para trabajar con niños. Se pueden realizar talleres de diseño de juguetes utilizando materiales reciclados, como botellas de plástico, latas o cajas. En ellos, las ideas surgen en el grupo de manera rápida y los niños disfrutan de la experiencia de realizar algo con sus propias manos.

Los prototipos obtenidos con esta técnica son prototipos de baja fidelidad. Se trata de bocetos y maquetas que ayudan a la conceptualización y a la comunicación entre los miembros del equipo, permitiendo probar distintas ideas de diseño.

La siguiente figura muestra ejemplos del uso de la técnica *rough prototyping*.

² Fuente: <http://bluefrogcreativenetwork.com/sketching.php>



Figura 3: Ejemplos de *rough prototyping*³.

Affinity diagram

Esta técnica se utiliza para recopilar y organizar grandes cantidades de datos, ideas y puntos de vista utilizando sus correlaciones naturales. Concretamente, los diagramas de afinidad se deben utilizar cuando (1) el problema a resolver es complejo o difícil de entender, (2) el problema parece estar desorganizado, (3) el problema requiere de la participación y soporte de todo el equipo de trabajo, (4) se quiere determinar los conceptos claves dentro de un gran número de ideas.

El proceso seguido para realizar un diagrama de afinidad comienza por establecer el problema. Suele ser de gran ayuda determinar el problema en forma de pregunta.

A continuación, cada participante debe pensar sus ideas para solucionar el problema propuesto y escribirlas en tarjetas o notas adhesivas. Para reunir las ideas, se puede realizar una sesión de *brainstorming* [7], u obtener las ideas a través de observación directa, entrevistas o cualquier otro material de referencia.

Las notas creadas se pueden pegar en una pared para obtener una visión global de las ideas propuestas. A continuación, los miembros del equipo mueven y organizan las notas en grupos

³ Fuentes: <http://www.servicedesigntools.org/tools/30>, <http://ahmetemreacar.info/tag/prototyping/>

basándose en las relaciones y asociaciones que establecen entre los distintos conceptos. Se busca la correlación entre el contenido de las notas, de manera que sea posible identificar grupos significativos de ideas para resolver el problema, como se muestra en la *Figura 4*. Las notas deberán leerse y revisarse de nuevo para verificar que se han agrupado de forma adecuada.

El resultado es una especie de representación verbal y visual que describe el primer esbozo de la solución de diseño. Para generar buenas soluciones, es importante tener en cuenta las siguientes prácticas:

- Los miembros del equipo deben exponer cualquier idea que se les ocurra, sin importar de que tipo sea o si parecen ambiguas o inaplicables.
- Los participantes no deben discutir ninguna de las ideas expuestas, ya que se podría inhibir la opinión de otros participantes. Igualmente, las ideas no deben ser criticadas ni juzgadas.
- Sólo se deben realizar preguntas para aclarar conceptos. Es necesario que el resto de los miembros del equipo opinen.

Una vez se han presentado y agrupado las ideas, es necesario asignar un título a cada uno de los grupos por medio de una discusión en grupo. Estos títulos deben describir de manera concisa lo que el grupo representa. Se deben pensar como si fuesen titulares periodísticos, y no deberían ser palabras sueltas. De esta manera, el grupo debería resultar significativo independientemente del contenido de los demás grupos. Si es necesario, se pueden utilizar títulos para crear subgrupos.

Este proceso se repite hasta que todos los grupos tengan un nombre. Cualquier nota individual que no se encuentre agrupada puede incluirse en un grupo de “Misceláneos”. Para terminar, el grupo de trabajo debe discutir la relación de los grupos y sus elementos correspondientes con el problema. Para ello, puede crearse una matriz en la que se enfrente cada solución obtenida a través de los grupos con los objetivos del problema planteado.

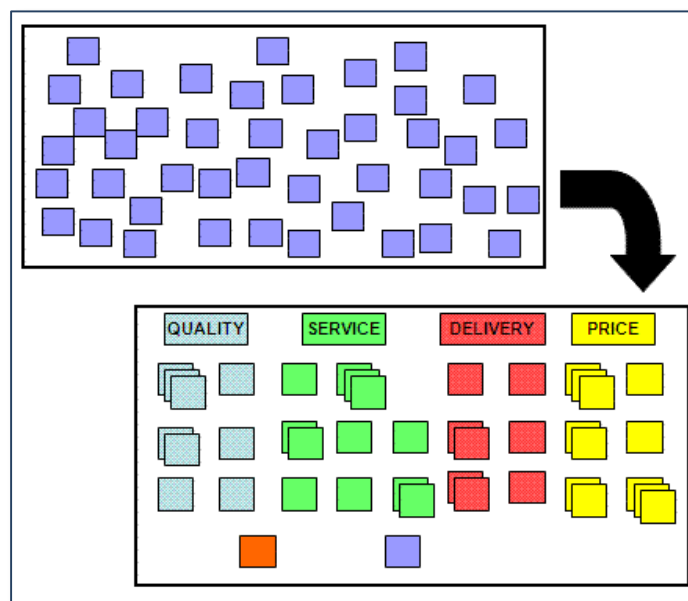


Figura 4: Affinity diagram⁴.

Motivation matrix

Esta técnica permite la investigación de una solución asumiendo el punto de vista de cada actor con sus propios intereses [8].

El objetivo de la matriz de motivación es la comprensión de las conexiones entre los diferentes actores del sistema, o *stakeholders*. Esto es posible gracias a la obtención de la motivación que cada uno de ellos tiene durante su participación en el sistema. Cada actor expresa lo que necesita o espera del sistema. En concreto, la creación de una matriz de motivación permite responder a las siguientes preguntas:

- ¿Por qué los actores están involucrados en la solución?
- ¿Por qué los actores son considerados como grupos de interés, o *stakeholders*?
- ¿Cuáles son sus intereses?

El proceso de desarrollo de esta técnica se basa en el despliegue de un *checklist* de motivaciones, beneficios y contribuciones de cada punto de vista de cada actor. Las referencias cruzadas de los actores permiten comprobar lo que cada *stakeholder* puede aportar y lo que cada uno puede obtener del sistema, como se muestra en la Figura 5.

⁴ Fuente: <http://www.six-sigma-material.com/Affinity-Diagram.html>

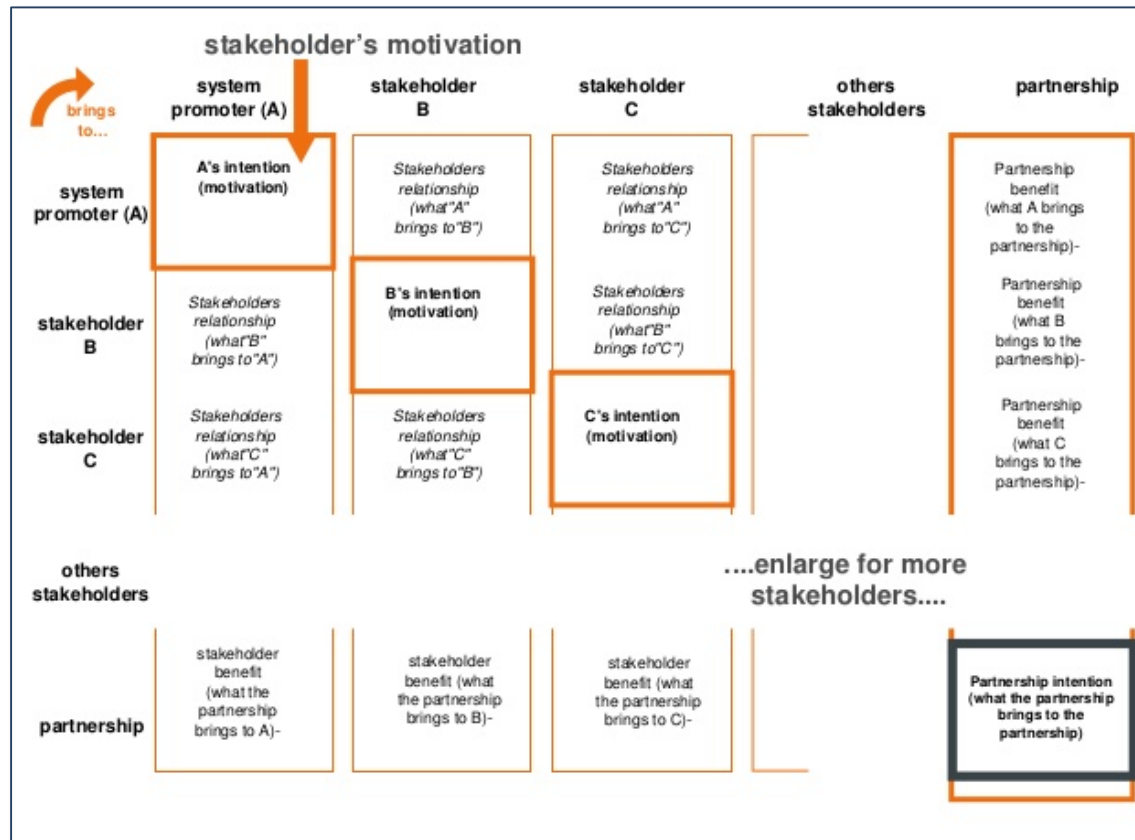


Figura 5: Referencias cruzadas en una *motivation matrix*⁵.

Mind map

Técnica para la estimulación visual de nuestros pensamientos y de sus conexiones. Esta técnica muchas aplicaciones en la vida personal, educativa y laboral. Una de las más importantes es la lluvia de ideas, ya que las ideas se insertan en el mapa radial alrededor del nodo central sin la priorización implícita que proviene de una jerarquía o de un sistema secuencial. La falta de jerarquía determinada por el desarrollo radial del concepto inicial hace que sea una herramienta realmente útil en sesiones de reflexión en las que varias personas pueden contribuir al desarrollo de una misma idea. De esta manera, también puede utilizarse para encontrar el centro real del problema a resolver.

La visualización comienza con un problema o una idea, que se sitúa en el centro del mapa. A partir de esta idea, se utilizan símbolos, líneas, palabras y dibujos para construir un sistema de pensamientos alrededor del punto de partida.

⁵ Fuente: <http://www.slideshare.net/urijoe/visualization-tool-how-communicate-the-service-design-concepts-presentation>

Por tanto, algunos de los usos de los mapas mentales son:

- Resolución de problemas dentro de un contexto de trabajo.
- Creación de un esquema o marco de diseño.
- Colaboración anónima en el desarrollo de ideas.
- Expresión individual y grupal de la creatividad.
- Condensación de un conjunto de información en un formato conciso y memorizable.

Los mapas mentales están muy relacionados con los mapas conceptuales. Sin embargo, la creación de mapas conceptuales exige el seguimiento de unas fases de desarrollo, mientras que la creación de mapas mentales es un proceso más relajado y desordenado de expresión de ideas. Por otra parte, los mapas mentales se centran en el desarrollo de una sola idea central, mientras que los mapas conceptuales se utilizan para conectar múltiples ideas.

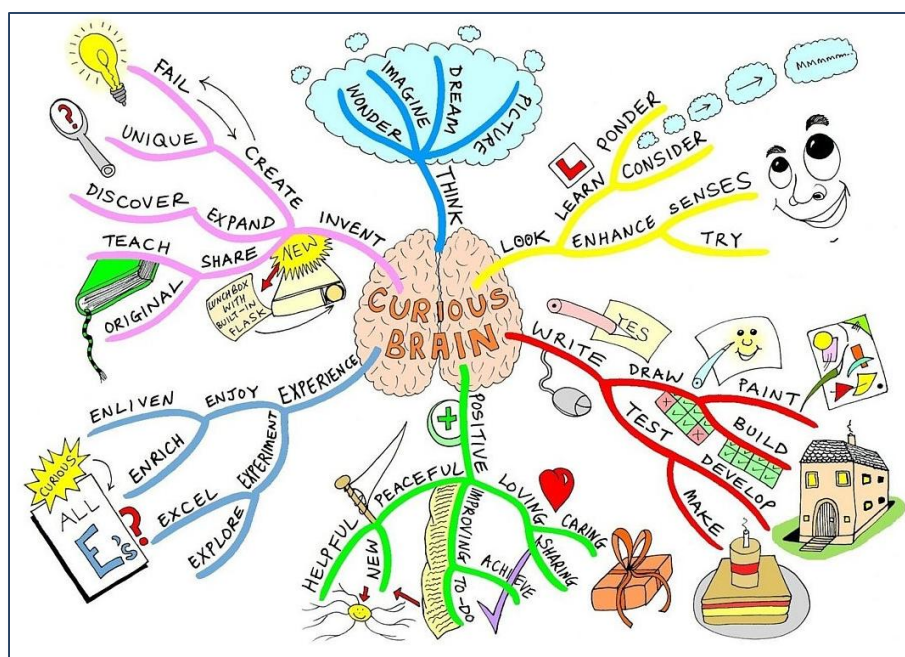


Figura 6: Mind map⁶.

Storyboard

Un *storyboard* es una representación de casos de uso a través de una serie de dibujos o imágenes que forman una secuencia narrativa. Proviene de la tradición cinematográfica, en la que

⁶ Fuente: www.mindmapinspiration.com

un conjunto de imágenes mostradas en secuencia sirven de guía para entender una historia, previsualizar una animación o seguir la estructura de una película antes de filmarse.

La creación de *storyboard* fomenta la colaboración, pues el proceso de pensamiento visual y la planificación permite que un equipo de trabajo proponga sus ideas y genera consenso dentro del grupo.

El proceso de creación de un *storyboard* se puede realizar de manera colaborativa de varias maneras. Una forma sería que cada miembro del grupo de trabajo diseñase unas de las viñetas del *storyboard* para después establecer la secuencia de viñetas entre todos los miembros del equipo. Otra manera supondría que se estableciese un orden por el que cada miembro introduce una nueva viñeta a partir de la anterior.

Es una técnica muy utilizada en el desarrollo de software como parte de la identificación de las especificaciones de un software particular. Durante la fase de especificación, se crea la sucesión de pantallas que mostrará el software, ya sea en papel o a partir de un software especializado, para ilustrar los pasos importantes de la experiencia interactiva del usuario (véase [Figura 7](#)). El *storyboard* es modificado por los diseñadores y los clientes al mismo tiempo que se extraen nuevas necesidades a cubrir por el software. De esta manera, ayuda al usuario a comprender exactamente cómo funcionará el software antes de la creación del mismo. Por otra parte, resulta más económico realizar cambios sobre un *storyboard* que sobre una aplicación software ya implementada.

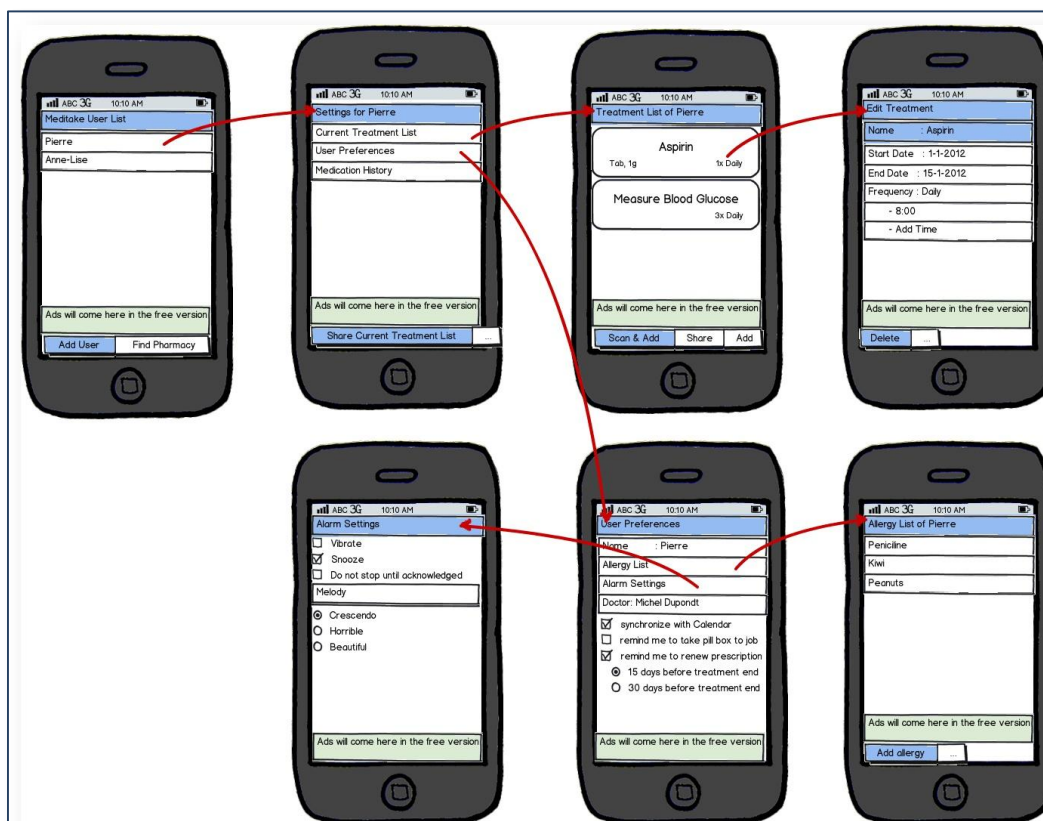


Figura 7: Storyboard de una aplicación móvil⁷.

Storytelling

En el *storytelling* la narración se utiliza para ilustrar la solución a un problema a través del uso de palabras simples. De esta manera, se puede comunicar una idea dentro de un grupo de trabajo, y preparar el primer esbozo a la solución del problema.

Se suelen dejar huecos en blanco en la narración, de manera que sean rellenados a partir de las sugerencias de otros miembros del grupo, otras partes interesadas, o por los propios usuarios.

El *storytelling* se utiliza en el campo del *marketing* y la publicidad. Su objetivo no es persuadir hablando de las características de un producto para captar usuarios, sino que el objetivo es conectar con la parte emocional del usuario.

Es una técnica muy utilizada en el desarrollo de software como forma de establecer una forma de comunicación de ideas. Para ello, es aconsejable comenzar el *storytelling* con fragmentos de historias o anécdotas breves, con el fin de interactuar con los usuarios. De cada una de las

⁷ Fuente: <https://meditake.atlassian.net/wiki/display/WEBM/Welcome+to+MediTake>

anécdotas seleccionadas, es necesario identificar el contexto, las emociones y los elementos específicos de la historia. Es necesario plantear el *storytelling* como la narración de una historia, en la que es preciso identificar las escenas o secciones de la historia, con una anécdota en cada una.

Elección de una técnica de co-diseño

Tras realizar el análisis teórico de las técnicas de co-diseño más relevantes, es necesario escoger la técnica de co-diseño que completará a la antigua aplicación ColDes y que permitirá convertir a ésta en una aplicación que asista el co-diseño.

Para ello, la técnica escogida debe cumplir las siguientes restricciones o requisitos:

1. Posibilidad de ser implementada en una aplicación web.
2. Permitir la asistencia a la identificación de un problema y a la provisión de soluciones.
3. Permitir la asistencia al desarrollo software en todas sus fases.

La primera de las características se cumple para todas las técnicas descritas, excepto para *rough prototyping*, por lo que se descarta esta técnica.

En cuanto a la segunda técnica, no sería válido el uso de matrices de motivación, ya que parte de la solución a un problema dado. Tampoco sería muy adecuado utilizar *storyboard*, ya que es necesario partir de la identificación de las especificaciones de un software particular.

Tras realizar una primera eliminación de las técnicas menos adecuadas, se obtienen las técnicas *group sketching*, *affinity diagram*, *mind map* y *story telling*. Entre ellas, es necesario escoger aquella más flexible, que permita asistir todas las fases de desarrollo software: análisis, diseño y evaluación. Mientras que *mind map* y *story telling* resultan más adecuadas para asistir la fase de análisis y *group sketching* resulta más adecuada para asistir el proceso de diseño, la técnica *affinity diagram* permite asistir las tres fases del proceso de desarrollo software.

En la [Figura 8](#) se presenta un diagrama en el que se muestran cuáles de las técnicas de co-diseño cumplen cada uno de los requisitos impuestos líneas arriba. Como se puede observar en la figura, y tal y como se dedujo en los párrafos anteriores, la única técnica que cumple con los tres requisitos es *affinity diagram*, por lo que es la técnica escogida para convertir a ColDes en una herramienta que asista las actividades de co-diseño.

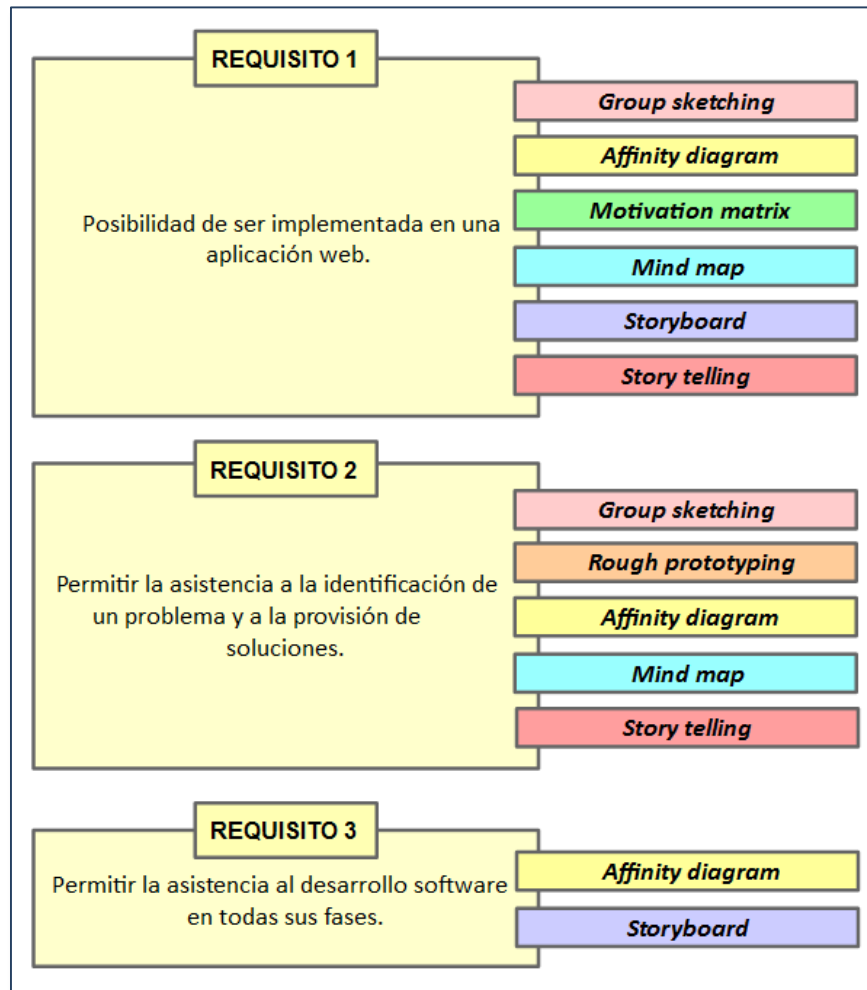


Figura 8: Técnicas de co-diseño según los requisitos del proyecto.

2.2 Revisión de tecnologías móviles

Tal y como se define en el capítulo de introducción, el objetivo del presente proyecto es el de extender la funcionalidad de la herramienta colaborativa ColDes para convertirla en una herramienta que asista el proceso de co-diseño. Con este propósito, se busca desarrollar una aplicación móvil que permita al usuario gestionar recursos multimedia que puedan ser utilizados como elementos de co-diseño tanto de manera co-localizada como distribuida.

De acuerdo a esto, a lo largo de esta sección se realiza una revisión de distintas tecnologías móviles. En particular, se han escogido las siguientes: **Android**, **iOS**, **Windows Phone**, **JavaFx**, y **jQuery**. Como criterios definidos para su revisión se establecen su relevancia en el mercado actual de *smartphones* y *tablets* y su carácter multiplataforma, el cuál posibilita la ejecución de aplicaciones en cualquier navegador web.

A continuación, se definen los criterios de estudio que servirán para realizar la revisión teórica, así como su relevancia en el proceso de elección de una de las tecnologías. Seguidamente, se presentará la información más importante de cada criterio para cada tecnología en una tabla comparativa. A partir de esta información, se realizará una comparativa textual de las tecnologías, definiendo los pros y contras de cada una de ellas. Finalmente, se realizará una comparación ponderada de las cinco tecnologías y se escogerá una de ellas para el desarrollo de la aplicación móvil de ColDes.

Criterios de evaluación

Antes de comenzar la revisión teórica de las tecnologías, es necesario establecer unos criterios o características relevantes que serán analizados para cada tecnología, y que permitirán realizar posteriormente una comparación de las tecnologías. Los criterios que se analizarán para cada una de las tecnologías son los siguientes:

Curva de aprendizaje:	Grado de dificultad a la hora de aprender la tecnología y porqué. Este criterio cuenta con un gran componente personal y se considera de gran importancia para escoger una tecnología, ya que se va a desarrollar un proyecto en un tiempo limitado.
Coste:	Coste de uso y desarrollo de la tecnología. Al tratarse de un proyecto universitario, también se considera de gran importancia este criterio.
Comunidad de usuarios y desarrolladores:	Alcance de la tecnología a partir del número de usuarios y desarrolladores. Se considera un criterio de gran importancia en la comparación de tecnologías, ya que el desarrollo de una aplicación puede llevar implícita su salida al mercado.
Rendimiento:	Análisis de velocidad, capacidad de procesamiento, fluidez, tiempos de ejecución...
Documentación:	Cantidad de información disponible sobre la tecnología y el lenguaje de programación. Se considera un criterio de gran importancia en la comparación de tecnologías, ya que desarrollar una aplicación para una tecnología poco documentada implica una dedicación de tiempo mucho mayor que para una tecnología para la que existe mucha documentación.

A continuación, y antes de analizar cada uno de los criterios definidos anteriormente, se presenta en la [Tabla 1](#) el resumen del análisis teórico realizado para cada una de las tecnologías propuestas.

	Android	iOS	JavaFx Mobile	jQuery Mobile	Windows Phone
Curva de Aprendizaje	Lenguaje Java. Herramientas para novatos.	Lenguaje propio (Objective-C).	Java y lenguajes web (XML, CSS).	Lenguajes web (Javascript, HTML, CSS)	Programación orientada a objetos. C#
Coste	Entorno multiplataforma. Licencia de publicación \$25.	Entorno <i>Mac</i> . Licencia de publicación \$99 al año.	Entorno multiplataforma. Licencias gratuitas.	Entorno multiplataforma. Licencias gratuitas.	Entorno <i>Windows</i> . Licencia de publicación \$99 al año.
Comunidad de Usuarios	Gran popularidad.	Gran popularidad.	En decrecimiento.	En decrecimiento.	En crecimiento.
Comunidad de Desarrolladores	Muy abundante.	Muy abundante.	En decrecimiento.	Abundante sólo en <i>Github</i> .	En crecimiento. Ayudas de <i>Microsoft</i> para el desarrollo.
Rendimiento	Medio (Máquina virtual de Java).	Excelente.	Malo (Máquina virtual de Java + navegador web).	Malo.	Bueno.
Documentación	Mucha información.	Mucha información.	Página oficial.	Página oficial.	Página de <i>Microsoft</i> y en páginas externas.

Tabla 1: Resumen del análisis teórico de las tecnologías.

Evaluación cualitativa

Tras presentar en la [Tabla 1](#) el resumen del análisis teórico de las tecnologías propuestas, se va a realizar una evaluación textual de las características de cada tecnología. Esta evaluación consiste en presentar los pros y contras de cada tecnología y ponderar cada uno de los criterios analizados con el fin de escoger finalmente la tecnología más adecuada.

La popularidad de las tecnologías en auge o con un futuro prometedor, en el que habrá una gran comunidad de usuarios, es un gran incentivo a tener en cuenta. Este criterio lo satisfacen los sistemas operativos Android, iOS y, en menor medida, Windows Phone. Gracias a una gran estrategia de expansión de Windows, en colaboración con otra gran empresa como es Nokia, se puede considerar a Windows Phone como potencial competidor en relación a la comunidad de usuarios de un sistema operativo móvil.

Por otro lado, también se ha considerado **la curva de aprendizaje** como uno de los criterios más importantes. Dado el carácter de este proyecto, supone una restricción el aprender un lenguaje en un corto periodo de tiempo e implementar el prototipo de una aplicación móvil. En este punto, se consideran como potenciales tecnologías Android, JQuery Mobile y Windows Phone. Las dos primeras utilizan lenguaje Java o Javascript, del que hay infinidad de información

para su desarrollo. El tercero, Windows Phone, está programado en C#, una variante de C orientado a objetos, del cual se dispone de mucha información, tutoriales, ejemplos, etc. En cuanto a iOS, utiliza un lenguaje único, cerrado y con una gran curva de aprendizaje, sin ninguna similitud a los lenguajes más habituales. En cuanto a JavaFx, utiliza un lenguaje híbrido entre XML, Java y CSS, sin apenas documentación inicial.

De las cinco tecnologías propuestas, JavaFX y JQuery Mobile están orientados, en mayor medida, a la creación de aplicaciones web, lo que implica complicaciones para el uso de características propias de los dispositivos móviles, como la función de *multitouch*, el reconocimiento de voz o el uso de la cámara.

El tercer aspecto y el último que se ha ponderado como una de los más determinantes es el **coste**. Las tecnologías más interesantes en este aspecto son Android, JavaFx y JQuery. Tanto para Windows Phone como para iOS, la licencia para la publicación de aplicaciones supone \$99 anuales. Además, son necesarios dispositivos propios de las marcas para el desarrollo de aplicaciones, es decir, que se necesitará un ordenador *Mac* para programar para iOS y un entorno *Windows* para programar aplicaciones para Windows Phone. Esto encarece aun más la puesta en marcha del desarrollo de aplicaciones para estas plataformas.

La comunidad de desarrolladores arroja una tecnología y le proporciona más vida útil a dicho sistema operativo, lo que garantiza unos años mínimos de funcionamiento. En este criterio pasan la prueba las tecnologías Android, iOS y Windows Phone. Por otra parte, se encuentra la cantidad de documentación existente para la tecnología, que actualmente está ligada al éxito del sistema operativo y al número de comunidades de usuarios y desarrolladores existentes. De nuevo, Android, iOS y Windows Phone obtienen la mejor nota para este criterio. JavaFx y JQuery no cuentan con mucha documentación fuera de las páginas oficiales, foros y algún libro.

Por último, el rendimiento es otra característica importante. Las máquinas virtuales Java no tienen fama de ser rápidas, por lo que desecharemos a Android, JavaFx y JQuery. Las Windows Phone y iOS se consideran mejores en comparación con las anteriores.

Con el fin de completar la información necesaria para escoger una de las tecnologías, se presentan a continuación los pros y contras de cada una de las tecnologías:

Android

Pros:

- El SDK es multiplataforma.
- El desarrollo es gratuito, y la licencia de desarrollador es vitalicia.
- Facilidad de ejecutar aplicaciones de terceros en un dispositivo Android.
- Facilidad de publicar aplicaciones propias en *Google Play*.

Contras:

- El rendimiento de Java.
- La fragmentación. Existen muchos dispositivos Android distintos, con diferentes resoluciones de pantalla, diferentes procesadores, diferentes versiones del sistema operativo... Esto dificulta la tarea del desarrollador, que debe crear aplicaciones que puedan ser ejecutadas en el mayor número posible de dispositivos.

iOS

Pros:

- La respuesta a las órdenes del usuario es inmediata y provee de una interfaz fluida.
- La interfaz de usuario de iOS está basada en el concepto de manipulación directa, usando gestos multitáctiles.
- Apple lanza sus actualizaciones de manera oficial y casi cada tres meses.
- La gran comunidad de usuarios y desarrolladores de *Apple*.

Contras:

- iOS no permite *Adobe Flash* ni Java, usa HTML5 como una alternativa a *Flash*.
- El precio de los dispositivos con iOS lo limita a un público más determinado.

Windows Phone

Pros:

- Lenguaje orientado a objetos actual: C#. Fácil de leer, mantener y portable.
- Menos código que en otras plataformas para realizar la misma función.
- Se puede prototipar la interfaz gráfica y la aplicación en la misma herramienta.
- La herramienta Expression Blend para el diseño de la interfaz en Windows Phone es una herramienta muy sencilla de utilizar y más potente que en la mayoría de sus competidores en esta evaluación de tecnologías. [36]
- Microsoft ha potenciado al máximo las herramientas de desarrollo de Windows Phone 7, lo que se traduce en una programación sencilla y más rápida que para otras plataformas.

Contras:

- Necesario tener sistema operativo Windows Vista o Windows 7.
- Aún no demasiado extendido entre la comunidad de usuarios.
- Aún no disponible en muchos móviles.
- Mercado de aplicaciones aún en crecimiento, sin tantas aplicaciones como en Android o iOS.

jQuery Mobile

Pros:

- Fácil de aprender y rápido de desarrollar.
- Multiplataforma.
- Se evita la aprobación de las tiendas de aplicaciones inicialmente y a cada actualización del software.

Contras:

- Más lento que las aplicaciones nativas.
- Capacidades limitadas frente a nativas. Las acciones dadas por *JavaScript*, que se ejecuta en un navegador, no pueden tener acceso completo a todas las características del dispositivo, como la cámara.

JavaFX

Pros:

- La mayoría de las aplicaciones de JavaFX pueden ejecutarse en ordenadores y en dispositivos móviles.

Contras:

- No está demostrado que sea compatible con la tecnología Android o iOS.
- No tiene una tienda de aplicaciones donde compartir tus creaciones.
- No compite con tecnologías móviles en el mercado sino con *Adobe Flex* o *Silverlight*, por lo que está más orientada a la web que a los dispositivos móviles. Actualmente están más centrados en competir con HTML5.

Evaluación cuantitativa

	Curva de Aprendizaje	Coste	Comunidad de Usuarios
Android	2	2	2
iOS	0	0	2
JavaFx Mobile	1	2	0
jQuery Mobile	2	2	0
Windows Phone	2	0	1

Tabla 2: Comparación ponderada de las tecnologías.

Los criterios más relevantes para la elección de la tecnología móvil han sido la curva de aprendizaje, el coste y la comunidad de usuarios. De acuerdo a esto, a cada tecnología se le ha asignado, para cada uno de los criterios, una puntuación entre 0 y 2. Una puntuación de 2 en un criterio dado supone que la tecnología satisface completamente las necesidades del proyecto. De igual manera, una puntuación de 0 en un criterio dado supone que la tecnología no satisface las necesidades del proyecto. La puntuación de 1 se reserva para la satisfacción parcial de las necesidades del proyecto. En base a ello, la [Tabla 2](#) despliega la puntuación asignada a cada tecnología para cada criterio.

De acuerdo a la [Tabla 2](#) se puede observar que Android es la única tecnología que satisface completamente los criterios que se han considerado más relevantes. Cabe destacar que debido a las características del presente proyecto no se ha considerado el coste de publicación de aplicaciones Android como un coste significativo, por lo que se ha establecido la mayor puntuación para este criterio. Con todo, Android se escoge como la tecnología sobre la que se implementará la aplicación móvil para ColDes.

3 Gestión de proyecto software

En este apartado se realiza un supuesto de la gestión del proyecto software desarrollado. En concreto, esta gestión de proyecto hace referencia a la tarea de detallar las decisiones tomadas en cuanto a los recursos necesarios para el desarrollo del mismo. Con este fin, se ha optado por realizar la gestión de proyecto como si se tratase de la gestión de un proyecto real, realizado con las condiciones habituales del entorno empresarial. De acuerdo a ello, todas las estimaciones se realizan de manera hipotética, pudiéndose contar con todos los recursos humanos y materiales que se estimen necesarios para llevar a cabo el proyecto. Por otra parte, se busca la rentabilidad del proyecto a partir de la creación de un presupuesto.

A continuación, y en primer lugar, se detalla la estimación de recursos necesarios para la realización del proyecto, describiendo paso a paso los cálculos realizados. Posteriormente, con los resultados obtenidos, se estima el presupuesto total del proyecto.

3.1 Estimación de recursos

En las siguientes líneas se presentan los costes de los distintos recursos necesarios para la realización del proyecto. Los costes que se tienen en cuenta son los siguientes: costes de personal, costes de hardware y software, costes de material y costes de gastos operativos.

Para realizar esta estimación no se ha utilizado ningún método formal de estimación, como los puntos de función, sino que se ha utilizado la información dada por la experiencia. A lo largo de las asignaturas de la titulación se han realizado estimaciones de presupuesto y se han realizado varios proyectos software que sirven para estimar los tiempos de realización de un proyecto y los costes asociados. Basándose en esta experiencia, se establece la duración del proyecto en cuatro meses.

Costes de personal

Para la realización de este proyecto es indispensable contar con un equipo de trabajo. En este equipo de trabajo debe existir personal para desempeñar cada uno de los siguientes roles:

Jefe de proyecto:	Encargado de coordinar al equipo de trabajo y dirigir el proyecto. Para ello, debe realizar el seguimiento, control, revisión y evaluación de los resultados obtenidos.
Analista:	Responsable de elaborar un catálogo de requisitos que describa con precisión el sistema. A partir de ello, debe recomendar las opciones tecnológicas que permitan cumplir con los requisitos del proyecto.
Diseñador:	Encargado de diseñar las posibles soluciones al problema planteado a partir de las especificaciones planteadas por el analista.

Programador: Responsable de construir el código que da lugar al producto resultante en base a las especificaciones del diseñador. También es encargado de realizar las pruebas unitarias de la aplicación.

Debido a que no se trata de un proyecto de gran envergadura, no es necesario que una persona desempeñe un único rol. En este caso, se contará con un equipo de trabajo de cuatro personas para realizar el proyecto. Cada una de las personas ocupará el mismo rol durante el tiempo de duración del proyecto, a partir de la siguiente relación:

- Una persona tomará el rol de jefe de proyecto. No será necesario que trabaje durante una jornada laboral completa. Trabajaré durante tres horas al día durante la duración total del proyecto.
- Una persona tomará el rol de analista. No será necesario que trabaje durante una jornada laboral completa. Trabajaré durante cinco horas al día durante el primer mes de duración del proyecto.
- Una persona tomará los roles de diseñador y programador para el desarrollo de la aplicación web. Trabajaré durante una jornada laboral completa durante los tres últimos meses del proyecto, coincidiendo con la finalización del trabajo del analista.
- Una persona tomará los roles de diseñador y programador para el desarrollo de la aplicación móvil. Trabajaré durante una jornada laboral completa durante los tres últimos meses del proyecto, coincidiendo con la finalización del trabajo del analista.

En la *Tabla 3* se muestra el coste de una hora de trabajo para cada persona del equipo de trabajo siguiendo las siguientes consideraciones:

- La jornada laboral será de un máximo ocho horas diarias. La dedicación de horas concreta vendrá dada por el rol desempeñado por el empleado.
- El número de días que un empleado deberá acudir a su puesto de trabajo se fija en cinco días naturales en cada semana del año, salvo en periodos vacacionales y festivos aprobados en el calendario laboral oficial.
- El coste por hora asignado a cada uno de los empleados viene dado por la situación socio-económica actual. En base a la experiencia en proyectos anteriores el coste por hora trabajada para cada empleado sería mayor, pero se ha decidido ajustar estos valores a la actual crisis económica mundial.
- El cálculo del salario bruto mensual de cada empleado se realiza a partir de la siguiente relación entre el coste de trabajo por hora, el número de horas trabajadas en una jornada laboral y el número de días trabajados en el mes:

$$\text{Salario bruto mensual} = \frac{\text{coste}}{\text{hora}} \times \frac{\text{horas}}{\text{día}} \times \frac{\text{días}}{\text{mes}}$$

Empleado	Coste / hora	Horas/día	Días/mes	Salario bruto mensual
Jefe de proyecto	25 €	3	22	1.650 €
Analista	20 €	5	22	2.200 €
Programador/diseñador aplicación web	16 €	8	22	2.816 €
Programador/diseñador aplicación móvil	16 €	8	22	2.816 €
TOTAL				9.482 €

Tabla 3: Salario bruto mensual del equipo de trabajo.

Tras calcular el coste mensual de los miembros del equipo de trabajo, es necesario calcular el coste de cotización a la seguridad social según la Ley de Presupuestos Generales del Estado para el año 2013 [9]. El porcentaje de cotización a la seguridad social asumible por la empresa es del 23,6% sobre el salario bruto de cualquier persona física que se encuentre en estado de alta laboral. Aplicando esta normativa, y teniendo en cuenta los meses de trabajo de cada empleado en el proyecto, se han obtenido los siguientes costes totales por persona:

Empleado	Salario bruto mensual	Cuota seguridad social	Meses de trabajo	Coste total
Jefe de proyecto	1.650 €	389,40 €	4	8.157,60 €
Analista	2.200 €	519,20 €	1	2.719,20 €
Programador/diseñador aplicación escritorio	2.816 €	664,58 €	3	10.441,73 €
Programador/diseñador aplicación móvil	2.816 €	664,58 €	3	10.441,73 €
TOTAL				31.760,26 €

Tabla 4: Coste total de personal.

Costes de hardware y software

En esta sección se presentan los costes asociados a los recursos software y hardware necesarios para el desarrollo del proyecto.

Los recursos hardware que se precisan en el desarrollo del proyecto son los siguientes:

- Ordenadores personales para cada uno de los miembros del equipo de trabajo. Se ha escogido el modelo de ordenador de sobremesa *HP Pavilion p6-2307es*, que incorpora teclado y ratón.
- Un monitor para cada uno de los ordenadores. Se ha escogido el modelo *HP Pavilion 22xi*.
- Una impresora multifunción que permita imprimir la documentación del proyecto. Se ha escogido el modelo *HP Officejet 4620*.
- Alquiler de un servidor virtual privado (VPS) en el que alojar el repositorio de código y la documentación del proyecto. Se ha escogido el alquiler de un VPS con sistema operativo *Ubuntu 10.4 LTS* en *1and1.es* [10].
- Dispositivo móvil con sistema operativo *Android* para la realización de las pruebas de la aplicación móvil. Se ha escogido un dispositivo *Samsung Galaxy SIII*.

Los recursos software necesarios para el desarrollo del proyecto que suponen un coste son los siguientes:

- Entorno de desarrollo *Adobe Flash Builder 4.5* para el desarrollo de la aplicación web.
- Suite de oficina para la generación de la documentación del proyecto. Se ha escogido la versión *Home and Business* de *Microsoft Office 2010*.

En la [Tabla 5](#) se muestra el coste asociado a recursos software y hardware según la siguiente fórmula:

$$\text{Coste imputable del recurso} = \frac{\text{meses de utilización}}{\text{periodo de depreciación}} \times \text{coste del recurso (sin IVA)}$$

En este caso, se estima que todos los recursos se utilizan durante los seis meses de duración del proyecto y cuentan con un periodo de depreciación de cuatro años (48 meses).

	Recurso	Coste (€)	Unidades	Coste total	Coste imputable
Hardware	Ordenador <i>HP Pavilion p6-2307es</i>	631,21 €	4	2.524,84 €	210,40 €
	Monitor <i>HP Pavilion 22xi</i>	125,60 €	4	502,40 €	41,87 €
	Impresora <i>HP Officejet 4620</i>	86,82 €	1	86,82 €	7,24 €
	Servidor <i>VPS Ubuntu 10.04 LTS</i>	370 €	1	369,78 €	30,82 €
	Dispositivo móvil <i>Samsung Galaxy SIII</i>	362,61 €	1	362,61 €	30,22 €
Software	<i>Adobe Flash Builder 4.5</i>	519,00 €	1	519,00 €	43,25 €
	<i>Microsoft Office 2010 Home and Business</i>	76,80 €	4	307,20 €	25,60 €
				TOTAL	389,39 €

Tabla 5: Coste imputable de recursos hardware y software.

Gastos generales

A parte de los gastos de personal y gastos asociados a recursos hardware y software, es necesario contemplar otros gastos generales, como los gastos de infraestructuras y mantenimiento.

Se debe tener en cuenta la contratación de la red eléctrica y de la línea telefónica y de Internet durante los seis meses de duración del proyecto, así como los gastos de mantenimiento de las infraestructuras.

Descripción	Tarifa mensual	Coste total
Contratación red eléctrica	120 €	480,00 €
Contratación línea telefónica y de Internet	29,50 €	118,00 €
Contratación servicio de limpieza y mantenimiento de las instalaciones	180 €	720,00 €
TOTAL		1.318,00 €

Tabla 6: Coste de infraestructuras y comunicaciones.

3.2 Presupuesto

Tras realizar el desglose de los costes de personal, costes de recursos hardware y software y costes de infraestructuras y mantenimiento, el coste final asociado al proyecto se calcula en la *Tabla 7*.

Descripción	Tarifa mensual
Costes de personal	31.760,26 €
Costes de recursos hardware y software	389,39 €
Gastos generales	1.318,00 €
TOTAL	33.467,64 €

Tabla 7: Coste total asociado al proyecto.

Para el cálculo del presupuesto del proyecto se tendrá en cuenta el cálculo de beneficios en base a un margen de beneficios del 20%. Igualmente, se aplica un margen de riesgos del 20%.

PRESUPUESTO FINAL	
Concepto	Coste
Costes total	33.467,64 €
Riesgos (10%)	3.346,76 €
Beneficio (20%)	6.693,53 €
TOTAL (sin IVA)	43.507,94 €
IVA (21%)	9.136,67 €
TOTAL	52.644,60 €

Tabla 8: Coste final asociado al proyecto.

Para facilitar el pago de los costes del proyecto, se establecen dos plazos de pago. El primero de ellos se realizará al inicio del proyecto, y el segundo al final del proyecto, según los importes que se muestran en la *Tabla 9*.

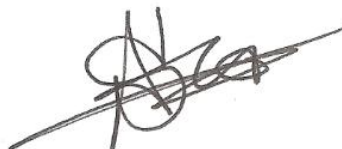
Plazos de pago	Fecha	% del coste total	Coste (sin IVA)	Coste total
Primer plazo	Inicio del proyecto	40%	17.403,17 €	21.057,84 €
Segundo plazo	Fin del proyecto	60%	26.104,76 €	31.586,76 €
			TOTAL	52.644,60 €

Tabla 9: Plazos de pago del coste final del proyecto.

De esta forma, **el presupuesto total del proyecto asciende a la cantidad de CINCUENTA Y DOS MIL SEISCIENTOS CUARENTA Y CUATRO CON SESENTA CÉNTIMOS (52.644,60 €).**

Leganés a 27 de mayo de 2003

El ingeniero proyectista

A handwritten signature in black ink, consisting of a stylized 'S' and 'L' followed by a long horizontal stroke.

Fdo. Sara Lucía Pérez García

4 Solución

En este apartado se describe la solución propuesta al problema expuesto en la sección *Problem definition*.

Concretamente, en primer lugar, se describe la fase de análisis, en la que se presenta una especificación detallada del sistema en forma de requisitos. En segundo lugar, se describe la fase de diseño, en la que se expone la arquitectura del sistema y las tecnologías utilizadas en el desarrollo. En esta fase también se incluye la descripción del proceso de prototipado de la interfaz del sistema. Por último, se describe la fase de implementación, presentando la organización del proyecto, así como las herramientas utilizadas para la implementación del sistema.

4.1 Descripción de la solución

Como se ha expuesto en apartados anteriores, el entorno desarrollado consta de dos partes diferenciadas: una aplicación web y una aplicación móvil. El entorno completo tiene como objetivo dar soporte a actividades de co-diseño. Este objetivo principal puede desglosarse en dos subobjetivos:

- 1) Realizar diagramas de afinidad de manera distribuida.
- 2) Recolectar información multimedia de manera contextual.

Para cumplir el primero de los subobjetivos se utiliza la antigua aplicación ColDes para crear un nuevo espacio de trabajo (equivalente a las antiguas salas) que asista el proceso de diseño.

Para cumplir el segundo de los subobjetivos, se desarrolla una aplicación móvil que permite obtener recursos multimedia en tiempo real. Estos recursos multimedia pueden ser utilizados en las actividades de co-diseño asistidas por ColDes. De esta manera, no es necesario disponer de todos los elementos de diseño en el lugar en el que se encuentra el espacio de diseño, sino que éstos se pueden obtener de espacios alejados del propio espacio de diseño.

Tras realizar esta introducción sobre el tipo de solución adoptada, se describe de manera más detallada cada una de las aplicaciones desarrolladas.

Aplicación web

Antes de comenzar a explicar las ampliaciones realizadas en ColDes para la adaptación de la aplicación a una herramienta que asista el proceso de co-diseño, es necesario hacer un breve resumen de la funcionalidad de la aplicación original (véase la sección 4.1 de la memoria del proyecto fin de carrera en el que se desarrolló ColDes [1]).

La motivación inicial a partir de la cual surge ColDes reside en la creación de una herramienta que permitiese la colaboración entre personas en entornos distribuidos. Para ello, ColDes se desarrolló como si se tratase de una pizarra colaborativa en la que distintos usuarios pueden realizar un proceso similar al de *group sketching*. Los usuarios pueden crear salas de trabajo en las que dibujar sus ideas y compartirlas con el resto de usuarios pertenecientes a la sala.

La colaboración dentro de una sala de trabajo se puede realizar de dos maneras:

- Por turnos: Los usuarios deben solicitar el pincel de dibujo para poder dibujar en la pizarra.
- Todos a la vez: Todos los usuarios pueden dibujar al mismo tiempo de manera síncrona.

Sin embargo, para completar el proceso de co-diseño, se incluirá en ColDes un nuevo tipo de salas de trabajo que permita realizar diagramas de afinidad. Estas salas se llamarán ahora *workspaces* y será necesario que el usuario elija el tipo de *workspace* antes de la creación del mismo.

Dentro de un *workspace* de diagrama de afinidad (véase [Figura 9](#)) los usuarios que formen parte del mismo podrán realizar el proceso completo de desarrollo de un diagrama de afinidad (véase sección [2.1](#)). Este proceso se puede realizar de dos modos:

1. Co-localizado: Cuando los usuarios realizan un diagrama de afinidad en el mismo espacio y lugar. Las ideas se pueden poner en común con el resto de miembros a través de la comunicación directa. Se trata del proceso habitual de realización de un diagrama de afinidad.
2. Distribuido: Cuando cada uno de los usuarios colabora en la creación del diagrama de afinidad desde un lugar diferente. Las ideas se pueden poner en común con el resto de miembros utilizando el chat habilitado en el *workspace*. En este caso, la colaboración dentro del *workspace* de diagrama de afinidad se puede realizar de la misma manera que la contemplada en las antiguas salas.

Para cualquiera de estos modos es posible realizar un diagrama de manera discontinua. Esto implica que en cualquier momento del proceso de creación de un diagrama de afinidad, cualquier usuario puede decidir guardar el estado actual del diagrama de afinidad para abrirlo más adelante y continuar trabajando en él.

Por otra parte, y dado que la creación de un diagrama de afinidad es un proceso iterativo, las primeras decisiones no tienen por qué ser las definitivas. Es por eso por lo que el usuario puede borrar ítems de información o deshacer grupos en cualquier momento.

A la hora de plasmar sus ideas en ítems de información, el usuario puede reforzar la información de esos ítems a través de los recursos multimedia recolectados a través de la aplicación móvil.

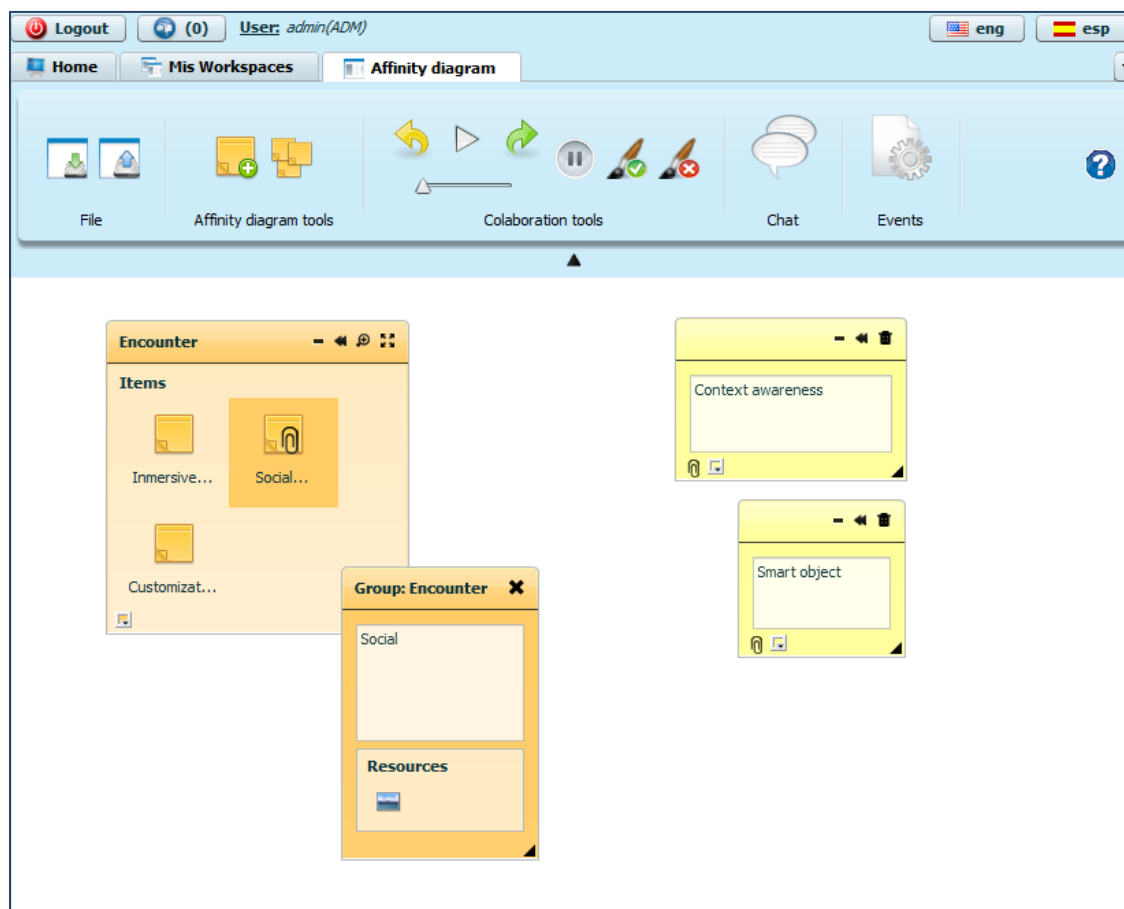


Figura 9: Aplicación web. *Workspace* para la creación de diagramas de afinidad.

Aplicación móvil

Los diagramas de afinidad se utilizan para diseñar elementos que tienen un vínculo con la realidad. Esto implica la existencia de elementos relacionados con el elemento de diseño. Es muy importante, por tanto, capturar todos los estos elementos e incorporarlos al diagrama de afinidad. Este es el objetivo de la aplicación móvil.

La aplicación móvil permite al usuario capturar sus ideas en cualquier momento y lugar para luego añadirlas a un diagrama de afinidad. Por ejemplo, el usuario puede fotografiar un escenario que le ha dado una nueva idea para incorporar en un diagrama de afinidad existente en la aplicación web. Igualmente puede capturar vídeos, audio, o puede añadir información textual a su zona de recursos.

Como ocurría en la aplicación web, y dado que el proceso de creación de un diagrama de afinidad no es definitivo, el usuario puede borrar en cualquier momento los recursos que había añadido a su cuenta.

Estos recursos pueden ser organizados dentro de la aplicación móvil, de manera que sea sencillo buscar en cualquier momento un recurso dado (véase [Figura 10](#)). Para ello, se utilizan etiquetas. Estas etiquetas son creadas en la aplicación y tienen asociado un color y texto único. De este modo, un usuario puede tener perfectamente organizados sus recursos en función de varios campos. Por ejemplo, podría tener organizados los recursos en función de los diagramas de afinidad en los que van a ser utilizados. En cualquier momento, el usuario puede visualizar el contenido de uno de los recursos añadidos a su cuenta y modificar las etiquetas asociadas al mismo.

Por tanto, la existencia de la aplicación móvil sólo tiene sentido al existir la aplicación web, ya que la primera ayuda a completar el proceso de co-diseño soportado por la segunda.

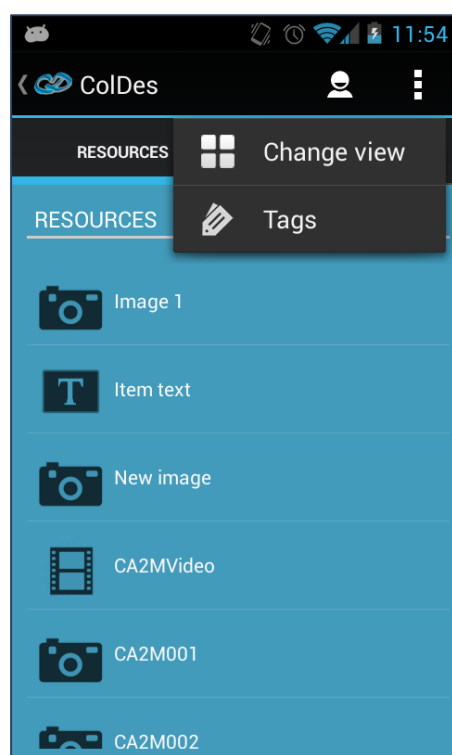


Figura 10: Aplicación móvil. Administración de recursos.

4.2 El proceso de desarrollo

En este apartado se detalla el proceso de desarrollo seguido para la resolución de un sistema que resuelva el problema descrito en la sección [Problem definition](#). En los siguientes apartados se

analiza cada una de las fases de desarrollo realizadas para la creación del sistema final: **análisis, diseño, implementación y pruebas**.

Modelo de proceso

El ciclo de vida de un proyecto software se define como un marco de referencia en el que se incluyen los procesos, actividades y tareas que forman parte del proceso de desarrollo, mantenimiento y despliegue de un producto software [11].

El modelo de ciclo de vida es el modelo que contempla el estado de las fases que componen un proyecto software, indicando el orden en el que se realiza cada una de las tareas y los criterios de transición entre ellas.

El modelo de proceso utilizado en el presente proyecto es un modelo de proceso iterativo e incremental [12]. Este modelo se basa en una serie de fases realimentadas y aplicadas repetidamente con una filosofía iterativa [13]. Las tareas de desarrollo se organizan separadamente en iteraciones, con el fin de que en cada una de ellas se logre una pieza de software construida sobre la versión lograda en iteraciones previas. El funcionamiento de un ciclo iterativo incremental se muestra en la *Figura 11*.

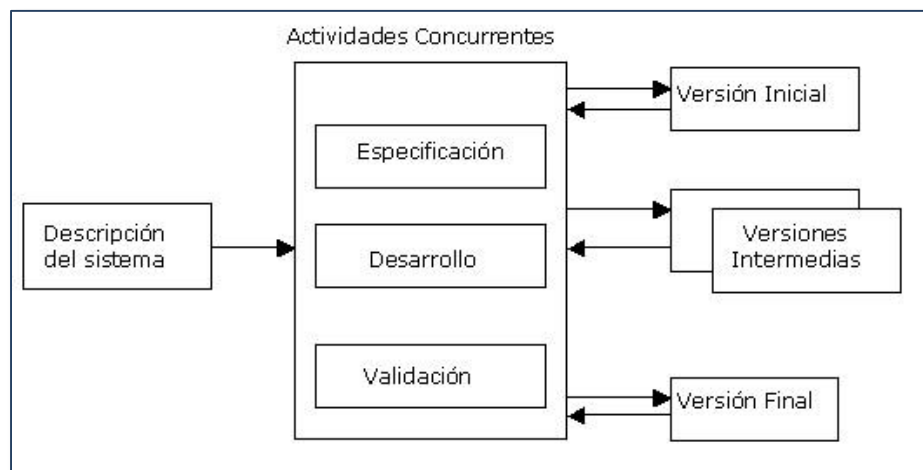


Figura 11: Esquema del funcionamiento de un ciclo iterativo incremental⁸.

Entre las ventajas del modelo de proceso iterativo e incremental se encuentran las siguientes:

- En cada iteración es posible obtener un producto parcial pero completamente operacional en lugar de una versión en la que se han reajustado los requisitos.

⁸ Fuente: https://commons.wikimedia.org/wiki/File:Modelo_Gral_Evolutivo_Incremental.jpg

- Se reduce la tasa de fallo del proyecto.
- Se mejora la productividad del equipo de trabajo a través de la experiencia obtenida iteración tras iteración.
- Permite manejar la complejidad del proyecto, apuntando a la resolución de los problemas por partes.

El motivo por el que el presente proyecto pertenece al modelo de proceso iterativo e incremental es porque se han realizado varios incrementos hasta llegar a completar una iteración. Igualmente, se han realizado varias iteraciones hasta obtener el producto final.

En concreto, en el proceso de análisis se han realizado dos incrementos. Como se verá en posteriores apartados, tras realizar el prototipado de la aplicación móvil en la fase de diseño, surgieron nuevos requisitos, que fueron añadidos al catálogo original. Tras agregar estos requisitos, se realizó un nuevo incremento en la fase de diseño, creando un nuevo prototipo que cumpliera con los nuevos requisitos. De igual manera, se realizó un segundo incremento de la fase de análisis al añadir nuevos requisitos tras realizar el segundo prototipo.

No obstante, el catálogo de requisitos presentado en este documento se corresponde con la versión final del mismo.

Análisis

La fase de análisis tiene por objetivo obtener una especificación detallada del sistema que se quiere construir para resolver el problema planteado. El análisis establece con precisión las necesidades y condiciones del sistema a desarrollar.

En los siguientes puntos se define la operativa del sistema, obtener los requisitos del sistema. En este proyecto concreto los requisitos vienen impuestos, y no es necesario reunirse con el cliente para extraer la información sobre el funcionamiento del sistema.

Tras definir los requisitos del sistema, se realizará la especificación de requisitos, donde se analizan y especifican los requisitos obtenidos anteriormente desde el punto de vista del comportamiento, estructura y funcionalidad del sistema.

Por último, se presentarán los casos de uso que recogen las necesidades definidas a partir de los requisitos.

Definición de requisitos

En esta fase de definición de requisitos, se presentan las funciones y restricciones que debe presentar el sistema a desarrollar. El conjunto de requisitos debe ser tan completo, constante y conciso como sea posible. Sin embargo, el primer catálogo de requisitos no tiene por qué ser

definitivo y está sujeto a modificaciones, pues el proceso de análisis exige continuas realimentaciones.

Se identifican dos tipos de requisitos: los requisitos funcionales y los requisitos no funcionales. Los requisitos funcionales definen el comportamiento interno del software y otras funcionalidades específicas que muestran cómo las necesidades serán llevadas a la práctica. Los requisitos no funcionales se enfocan al diseño o la implementación del sistema, y se utilizan para limitar los requisitos funcionales al imponer condiciones sobre los mismos.

A su vez, los requisitos presentados se dividen en los requisitos definidos para la aplicación web, y los requisitos definidos para la aplicación móvil.

Utilizando como base la plantilla de Especificación de Requisitos de *Volere* [14], se utiliza la siguiente plantilla para la especificación de requisitos:

Identificador	R<Tipo de requisito>-<Tipo de aplicación>-<Numero>		
Prioridad		Necesidad	
Descripción			
Prueba asociada			

Tabla 10: Plantilla de especificación de requisitos.

El significado de cada uno de los campos de la tabla anterior es el siguiente:

- **Identificador:** Identificador del requisito, compuesto por los siguientes campos:
 - **Tipo de requisito:** Diferencia entre requisito funcional y no funcional. Tomará el valor “F” si se trata de un requisito funcional, y “NF” si se trata de un requisito no funcional.
 - **Tipo de aplicación:** Diferencia entre los requisitos definidos para la aplicación web y los requisitos definidos para la aplicación móvil, tomando los valores “W” y “M” respectivamente.
 - **Número:** Número que identifica de manera unívoca el requisito.
- **Prioridad:** Establece el grado de importancia del cumplimiento de un requisito frente al resto. Puede tomar los valores Alta, Media o Baja.
- **Necesidad:** Establece el grado de exigencia del requisito. Puede tomar los valores Esencial, Deseable y Opcional.
- **Descripción:** Explica en qué consiste el requisito.

Se ha decidido modificar la plantilla de Volere original para ajustarla a las exigencias del proyecto. En primer lugar, se ha eliminado el campo que indica el tipo de requisito, ya que se especifica en el identificador del mismo. Igualmente, y dado que en este proyecto no se hace uso de casos de uso, se ha eliminado el campo que asocia un requisito con el caso de uso correspondiente.

Por otra parte, no se considera necesario especificar la justificación del requisito, ya que en este caso no existe un cliente del que obtener los requisitos, sino que éstos vienen impuestos por el objetivo del proyecto. Por el mismo motivo, no se ha incluido el campo que indica la persona que planteó el requisito ni los campos que indican la satisfacción del cliente.

Los campos de historia, materiales soportados y conflictos no se han incluido ya que se trata de un proyecto de pequeña envergadura en el que estos campos no aportarían gran información.

En su lugar, se ha incorporado el campo de necesidad, de manera que se pueda indicar cómo de necesario es el cumplimiento del requisito.

En esta sección de definición de requisitos sólo se muestra una definición breve del requisito. La especificación completa de los requisitos se encuentra en el [Anexo III. Especificación de requisitos](#).

Requisitos funcionales

Los requisitos funcionales que la aplicación web debe cumplir son los siguientes:

- RF-W-1:** Las salas de la antigua aplicación pasarán a llamarse *workspace*.
- RF-W-2:** El usuario podrá crear dos tipos de *workspace*: para la realización de *group sketching*, y para la creación de diagramas de afinidad. Los *workspace* para la realización de *group sketching* se corresponderán con las antiguas salas.
- RF-W-3:** Las opciones de creación de un *workspace* para la creación de diagramas de afinidad serán las mismas que para la creación de las antiguas salas de ColDes.
- RF-W-4:** Las opciones de colaboración de los *workspace* para la creación de diagramas de afinidad serán las mismas que en las antiguas salas de ColDes.
- RF-W-5:** En un *workspace* para la creación de diagramas de afinidad, el usuario podrá crear ítems de información.
- RF-W-6:** Todos los ítems de información creados por el usuario contendrán información textual.

- RF-W-7:** Cualquier usuario que pertenezca a un *workspace* para la creación de diagramas de afinidad podrá modificar el diagrama asociado al mismo.
- RF-W-8:** El usuario podrá modificar la posición de los ítems de información dentro del espacio de trabajo del *workspace*.
- RF-W-9:** El usuario podrá redimensionar los ítems de información dentro del espacio de trabajo del *workspace*.
- RF-W-10:** El usuario podrá elegir el color de los ítems de información.
- RF-W-11:** El usuario podrá eliminar los ítems creados anteriormente.
- RF-W-12:** El usuario podrá agrupar los ítems de información con contenido similar en grupos de ítems.
- RF-W-13:** El usuario podrá disolver un grupo de ítems de información.
- RF-W-14:** El usuario podrá dar un título a un grupo de ítems. Este título será identificativo para el grupo.
- RF-W-15:** El usuario podrá guardar el diagrama de afinidad asociado a un *workspace*.
- RF-W-16:** El usuario podrá cargar el diagrama de afinidad asociado a un *workspace*.
- RF-W-17:** El usuario podrá acceder a un espacio donde se encuentren los recursos externos.
- RF-W-18:** El usuario podrá asociar un recurso de sus recursos externos a un ítem de información.
- RF-W-19:** Los recursos añadidos a una nota adhesiva podrán ser de tipo audio, vídeo, imagen o texto.
- RF-W-20:** El usuario podrá eliminar el recurso asociado a una nota adhesiva.

Los requisitos funcionales que la aplicación móvil debe cumplir son los siguientes:

- RF-M-1:** Para hacer uso de la aplicación móvil, el usuario debe tener una cuenta en la aplicación web de ColDes.
- RF-M-2:** El acceso a la aplicación se realizará mediante un componente de autenticación que pedirá al usuario el nombre y la contraseña.
- RF-M-3:** El usuario podrá cambiar sus datos de usuario desde la aplicación móvil.
- RF-M-4:** El usuario podrá añadir recursos de audio, vídeo, imagen y/o texto a su cuenta.
- RF-M-5:** Al añadir un recurso, se pedirá al usuario el nombre del recurso y un comentario para el mismo.
- RF-M-6:** El usuario podrá obtener los recursos de imagen y vídeo desde la galería del dispositivo móvil o desde la cámara.
- RF-M-7:** El usuario podrá ver la lista de los recursos añadidos a su cuenta.
- RF-M-8:** El usuario podrá visualizar el contenido de cualquier recurso de la lista de recursos.
- RF-M-9:** El usuario podrá eliminar cualquier recurso de su lista de recursos.
- RF-M-10:** El usuario podrá crear etiquetas para asociarlas a los recursos.
- RF-M-11:** Al crear una etiqueta, se le pedirá al usuario el nombre de la etiqueta y el color asociado.
- RF-M-12:** El usuario podrá ver la lista de etiquetas añadidas a su cuenta.
- RF-M-13:** El usuario podrá asociar etiquetas de su lista a etiquetas a los recursos de su cuenta.
- RF-M-14:** El usuario podrá ver la lista de etiquetas asociadas a un recurso concreto.
- RF-M-15:** El usuario podrá ver la lista de recursos etiquetados por una etiqueta dada.
- RF-M-16:** El usuario podrá eliminar cualquier etiqueta creada anteriormente.

Requisitos no funcionales

Los requisitos no funcionales que la aplicación web debe cumplir se muestran a continuación. Éstos se suman a los que se especificaron para la aplicación original.

- RNF-W-1:** Un *workspace* para la creación de diagramas de afinidad permitirá crear un único diagrama de afinidad.
- RNF-W-2:** El aspecto de los ítems del diagrama de afinidad será similar al de una nota adhesiva.
- RNF-W-3:** El color de los ítems se escogerá a partir de un selector de colores incluido en el propio ítem.
- RNF-W-4:** Se permitirá añadir un único recurso a un ítem de información.

Los requisitos no funcionales que la aplicación móvil debe cumplir son los siguientes:

- RNF-M-1:** La aplicación deberá soportar los idiomas inglés y español.
- RNF-M-2:** La aplicación estará optimizada para su ejecución en dispositivos Android con una versión del sistema operativo 4.0 o superior.
- RNF-M-3:** La aplicación debe estar optimizada para las densidades de pantalla *mdpi*, *hdpi* y *xdpi*.
- RNF-M-4:** La conexión con la base de datos se realizará por medio de un *Web service* implementado en lenguaje Java.

Casos de uso

En esta fase de definición de casos de uso se presentan los casos de uso obtenidos a partir de la definición de requisitos realizada en las páginas anteriores. De manera general, un caso de uso no es más que una descripción de los pasos que el usuario debe realizar para llevar a cabo algún proceso [15]. En este caso concreto, sirven para definir los pasos que debe llevar a cabo el usuario para llegar a cumplir con la funcionalidad descrita en los requisitos del sistema. Estos pasos se definen a partir de la interacción entre el actor que inicia el caso de uso y el sistema.

Para presentar los casos de uso se utiliza la siguiente plantilla:

Caso de uso	Identificador	CU-<Tipo de aplicación>-<Numero>
Actores		
Descripción		
Precondiciones		
Curso básico		

Tabla 11: Plantilla de especificación de casos de uso.

El significado de cada uno de los campos [16] de la tabla anterior es el siguiente:

- **Caso de uso:** Nombre identificativo del caso de uso.
- **Identificador:** Identificador del caso de uso, compuesto por los siguientes campos:
 - **Tipo de aplicación:** Diferencia entre los casos de uso específicos para la aplicación web y los requisitos específicos para la aplicación móvil, tomando los valores “W” y “M” respectivamente.
 - **Número:** Número que identifica de manera unívoca el caso de uso.
- **Actores:** Entidad externa al sistema que demanda la funcionalidad dada por el caso de uso.
- **Escenario:** Explica en qué contexto se aplica el caso de uso.
- **Precondiciones:** Condiciones que se deben cumplir para que el curso de eventos pueda llevarse a cabo.
- **Curso básico:** Especifica la interacción entre los actores y el sistema.

A continuación, se presentan los casos de uso extraídos de los requisitos del sistema asociados a la aplicación web:

Caso de uso	Crear <i>workspace</i> para la creación de diagramas de afinidad	Identificador	CU-W-1
Actores	Usuario registrado		
Escenario	El usuario desea crear un nuevo <i>workspace</i> para la creación de diagramas de afinidad.		
Precondiciones	El usuario debe tener una cuenta en la aplicación web ColDes.		
Curso básico	<ol style="list-style-type: none"> 1. El usuario accede a la zona de <i>workspaces</i> de su cuenta. 2. El usuario accede a la zona de creación de <i>workspaces</i>. 3. El elige entre las opciones para crear un <i>workspace</i> la opción de crear de un <i>workspace</i> para la creación de diagramas de afinidad. 4. El usuario introduce la información relativa a la creación del <i>workspace</i>. 5. El usuario escoge el tipo de colaboración del <i>workspace</i>: “uno a uno” o “todos a la vez”. 6. El nuevo <i>workspace</i> es creado. 		

Caso de uso	Crear ítems de información	Identificador	CU-W-2
Actores	Usuario registrado		
Escenario	El usuario desea añadir notas de información en un diagrama de afinidad.		
Precondiciones	<ul style="list-style-type: none"> – El usuario debe tener una cuenta en la aplicación web ColDes. – El usuario debe pertenecer a un <i>workspace</i> para la creación de diagramas 		

	de afinidad.
Curso básico	<ol style="list-style-type: none"> 1. El usuario accede a la zona de <i>workspaces</i> de su cuenta. 2. El usuario accede al <i>workspace</i> para la creación de diagramas de afinidad en el que desea añadir nuevos ítems de información. 3. El usuario selecciona la acción de crear un nuevo ítem de información. 4. El usuario introduce el texto que acompaña al ítem de información. 5. Se crea el nuevo ítem de información con el texto introducido por el usuario.

Caso de uso	Personalizar ítems de información	Identificador	CU-W-3
Actores	Usuario registrado		
Escenario	El usuario desea añadir notas de información en un diagrama de afinidad.		
Precondiciones	<ul style="list-style-type: none"> – El usuario debe tener una cuenta en la aplicación web ColDes. – El usuario debe pertenecer a un <i>workspace</i> para la creación de diagramas de afinidad. – Debe existir algún ítem de información en el <i>workspace</i> para la creación de diagramas de afinidad. 		
Curso básico	<ol style="list-style-type: none"> 1. El usuario accede a la zona de <i>workspaces</i> de su cuenta. 2. El usuario accede al <i>workspace</i> para la creación de diagramas de afinidad en el que desea personalizar un ítem de información. 3. El usuario selecciona el ítem de información que desea personalizar. 4. El usuario redimensiona el ítem de información. 5. El usuario modifica el color del ítem de información. 6. El usuario cambia la posición del ítem de información. 7. El ítem adopta los cambios realizados por el usuario. 		

Caso de uso	Eliminar ítems de información	Identificador	CU-W-4
Actores	Usuario registrado		
Escenario	El usuario desea añadir notas de información en un diagrama de afinidad.		
Precondiciones	<ul style="list-style-type: none"> – El usuario debe tener una cuenta en la aplicación web ColDes. – El usuario debe pertenecer a un <i>workspace</i> para la creación de diagramas de afinidad. – Debe existir algún ítem de información en el <i>workspace</i> para la creación de diagramas de afinidad. 		
Curso básico	<ol style="list-style-type: none"> 1. El usuario accede a la zona de <i>workspaces</i> de su cuenta. 2. El usuario accede al <i>workspace</i> para la creación de diagramas de afinidad en el que desea eliminar un ítem de información. 		

	<ol style="list-style-type: none"> 3. El usuario selecciona el ítem de información que desea eliminar. 4. El usuario selecciona la opción de eliminar el ítem de información. 5. El ítem de información se elimina del diagrama de afinidad.
--	---

Caso de uso	Asociar recursos a un ítem de información	Identificador	CU-W-5
Actores	Usuario registrado		
Escenario	El usuario desea asociar recursos a un ítem de información.		
Precondiciones	<ul style="list-style-type: none"> – El usuario debe tener una cuenta en la aplicación web ColDes. – El usuario debe pertenecer a un <i>workspace</i> para la creación de diagramas de afinidad. – Debe existir algún ítem de información en el <i>workspace</i> para la creación de diagramas de afinidad. 		
Curso básico	<ol style="list-style-type: none"> 1. El usuario accede a la zona de <i>workspaces</i> de su cuenta. 2. El usuario accede al <i>workspace</i> para la creación de diagramas de afinidad. 3. El usuario selecciona la acción de añadir un nuevo recurso en el ítem de información escogido. 4. El usuario escoge el recurso que desea asociar al ítem desde la lista de recursos asociados a su cuenta. 5. El recurso se asocia al ítem de información. 		

Caso de uso	Eliminar recursos asociados a un ítem de información	Identificador	CU-W-6
Actores	Usuario registrado		
Escenario	El usuario desea eliminar recursos de un ítem de información.		
Precondiciones	<ul style="list-style-type: none"> – El usuario debe tener una cuenta en la aplicación web ColDes. – El usuario debe pertenecer a un <i>workspace</i> para la creación de diagramas de afinidad. – Debe existir algún ítem de información en el <i>workspace</i> para la creación de diagramas de afinidad al que se le hayan asociado recursos. 		
Curso básico	<ol style="list-style-type: none"> 1. El usuario accede a la zona de <i>workspaces</i> de su cuenta. 2. El usuario accede al <i>workspace</i> para la creación de diagramas de afinidad. 3. El usuario selecciona el ítem de información del que quiere eliminar un recurso. 4. El usuario selecciona el recurso asociado al ítem de información. 5. El usuario selecciona la opción de eliminar el recurso del ítem de información. 		

	6. El recurso se elimina del ítem de información.
--	---

Caso de uso	Agrupar ítems de información	Identificador	CU-W-7
Actores	Usuario registrado		
Escenario	El usuario desea agrupar varios ítems de información en un único grupo.		
Precondiciones	<ul style="list-style-type: none"> – El usuario debe tener una cuenta en la aplicación web ColDes. – El usuario debe pertenecer a un <i>workspace</i> para la creación de diagramas de afinidad. – Debe existir algún ítem de información en el <i>workspace</i> para la creación de diagramas de afinidad. 		
Curso básico	<ol style="list-style-type: none"> 1. El usuario accede a la zona de <i>workspaces</i> de su cuenta. 2. El usuario accede al <i>workspace</i> para la creación de diagramas de afinidad. 3. El usuario selecciona los ítems de información que desea agrupar. 4. El usuario selecciona la opción de agrupar ítems de información. 5. El usuario introduce el título del grupo. 6. Se crea un nuevo grupo que contiene los ítems de información seleccionados por el usuario. 		

Caso de uso	Disolver grupo de ítems de información	Identificador	CU-W-8
Actores	Usuario registrado		
Escenario	El usuario desea disolver un grupo de ítems creado anteriormente.		
Precondiciones	<ul style="list-style-type: none"> – El usuario debe tener una cuenta en la aplicación web ColDes. – El usuario debe pertenecer a un <i>workspace</i> para la creación de diagramas de afinidad. – Debe existir algún grupo de ítems de información en el <i>workspace</i> para la creación de diagramas de afinidad. 		
Curso básico	<ol style="list-style-type: none"> 1. El usuario accede a la zona de <i>workspaces</i> de su cuenta. 2. El usuario accede al <i>workspace</i> para la creación de diagramas de afinidad en el que desea disolver un grupo de ítems de información. 3. El usuario selecciona el grupo que desea disolver. 4. El usuario selecciona la opción de disolver el grupo de ítems. 5. El grupo de ítems desaparece, y aparecen los ítems que estaban asociados al mismo. 		

Caso de uso	Guardar diagrama de afinidad	Identificador	CU-W-9
Actores	Usuario registrado		
Escenario	El usuario desea guardar el estado actual del diagrama de afinidad.		
Precondiciones	<ul style="list-style-type: none"> – El usuario debe tener una cuenta en la aplicación web ColDes. – El usuario debe pertenecer a un <i>workspace</i> para la creación de diagramas de afinidad. 		
Curso básico	<ol style="list-style-type: none"> 1. El usuario accede a la zona de <i>workspaces</i> de su cuenta. 2. El usuario accede al <i>workspace</i> para la creación de diagramas de afinidad. 3. El usuario selecciona la opción de guardar el diagrama de afinidad. 4. Se guarda el estado actual del diagrama de afinidad. 		

Caso de uso	Cargar diagrama de afinidad	Identificador	CU-W-10
Actores	Usuario registrado		
Escenario	El usuario desea cargar un estado anterior del diagrama de afinidad.		
Precondiciones	<ul style="list-style-type: none"> – El usuario debe tener una cuenta en la aplicación web ColDes. – El usuario debe pertenecer a un <i>workspace</i> para la creación de diagramas de afinidad. 		
Curso básico	<ol style="list-style-type: none"> 1. El usuario accede a la zona de <i>workspaces</i> de su cuenta. 2. El usuario accede al <i>workspace</i> para la creación de diagramas de afinidad. 3. El usuario selecciona la opción de cargar el diagrama de afinidad. 4. Se carga el estado anterior del diagrama de afinidad. 		

Caso de uso	Visualizar recursos	Identificador	CU-W-11
Actores	Usuario registrado		
Escenario	El usuario desea visualizar un recurso añadido a partir de la aplicación móvil.		
Precondiciones	<ul style="list-style-type: none"> – El usuario debe tener una cuenta en la aplicación web ColDes. – El usuario debe haber añadido a su cuenta algún recurso a través de la aplicación móvil. 		
Curso básico	<ol style="list-style-type: none"> 1. El usuario accede a la zona de recursos de su cuenta. 2. El usuario escoge, de entre los recursos asociados a su cuenta, el recurso que desea visualizar. 3. El usuario visualiza el recurso. 		

A continuación, se presentan los casos de uso extraídos de los requisitos del sistema asociados a la aplicación móvil:

Caso de uso	Autenticación en el sistema	Identificador	CU-M-1
Actores	Usuario registrado		
Escenario	El usuario desea acceder a su cuenta mediante la aplicación móvil.		
Precondiciones	– El usuario debe tener una cuenta en la aplicación web ColDes.		
Curso básico	<ol style="list-style-type: none"> 1. El usuario introduce su nombre de usuario y su contraseña en el sistema. 2. El usuario accede al sistema. 		

Caso de uso	Modificar datos de usuario	Identificador	CU-M-2
Actores	Usuario registrado		
Escenario	El usuario desea modificar sus datos de usuario.		
Precondiciones	– El usuario debe tener una cuenta en la aplicación web ColDes.		
Curso básico	<ol style="list-style-type: none"> 1. El usuario accede a la zona de usuario. 2. El usuario modifica los datos de usuario que desea cambiar: nombre, apellidos, nombre de usuario, o contraseña. 3. Los datos de usuario se modifican. 		

Caso de uso	Añadir recursos	Identificador	CU-M-3
Actores	Usuario registrado		
Escenario	El usuario desea añadir un nuevo recurso a su cuenta.		
Precondiciones	– El usuario debe tener una cuenta en la aplicación web ColDes.		
Curso básico	<ol style="list-style-type: none"> 4. El usuario accede a la zona de recursos de la aplicación. 5. El usuario selecciona la opción de añadir un nuevo recurso. 6. El usuario escoge el tipo de recurso a añadir: imagen, vídeo, sonido o texto. 7. El usuario escoge el recurso a añadir de la galería del dispositivo o desde la cámara. 8. El usuario introduce el nombre, el comentario y las etiquetas asociadas al recurso. 9. El recurso se añade a la cuenta. 		

Caso de uso	Visualizar recursos	Identificador	CU-M-4
Actores	Usuario registrado		
Escenario	El usuario desea visualizar un recurso asociado su cuenta.		
Precondiciones	<ul style="list-style-type: none"> – El usuario debe tener una cuenta en la aplicación web ColDes. – El usuario debe haber añadido a su cuenta algún recurso. 		
Curso básico	<ol style="list-style-type: none"> 1. El usuario accede a la zona de recursos de la aplicación. 2. El usuario visualiza la lista de recursos asociados a su cuenta. 3. El usuario selecciona el recurso que desea visualizar. 4. Se muestra la información asociada al recurso: nombre, descripción y etiquetas asociadas. 5. Se visualiza el recurso. 		

Caso de uso	Eliminar recursos	Identificador	CU-M-5
Actores	Usuario registrado		
Escenario	El usuario desea eliminar un recurso de su cuenta.		
Precondiciones	<ul style="list-style-type: none"> – El usuario debe tener una cuenta en la aplicación web ColDes. – El usuario debe haber añadido a su cuenta algún recurso. 		
Curso básico	<ol style="list-style-type: none"> 1. El usuario accede a la zona de recursos de la aplicación. 2. El usuario visualiza la lista de recursos asociados a su cuenta. 3. El usuario selecciona el recurso que desea eliminar. 4. El usuario selecciona la opción de eliminar el recurso seleccionado. 5. El recurso se elimina de la cuenta del usuario. 		

Caso de uso	Añadir etiquetas	Identificador	CU-M-6
Actores	Usuario registrado		
Escenario	El usuario desea añadir una nueva etiqueta a su cuenta.		
Precondiciones	<ul style="list-style-type: none"> – El usuario debe tener una cuenta en la aplicación web ColDes. 		
Curso básico	<ol style="list-style-type: none"> 1. El usuario accede a la zona de etiquetas de la aplicación. 2. El usuario selecciona la opción de añadir una nueva etiqueta. 3. El usuario introduce el nombre y color de la etiqueta a crear. 4. La etiqueta se añade a su cuenta. 		

Caso de uso	Etiquetar recurso	Identificador	CU-M-7
Actores	Usuario registrado		
Escenario	El usuario desea etiquetar un recurso con una etiqueta de su cuenta.		

Precondiciones	<ul style="list-style-type: none"> – El usuario debe tener una cuenta en la aplicación web ColDes. – El usuario debe haber añadido a su cuenta algún recurso. – El usuario debe haber añadido a su cuenta alguna etiqueta.
Curso básico	<ol style="list-style-type: none"> 1. El usuario accede a la zona de recursos de la aplicación. 2. El usuario visualiza la lista de recursos asociados a su cuenta. 3. El usuario selecciona el recurso que desea etiquetar. 4. El usuario accede a la zona del recurso seleccionado. 5. El usuario selecciona la opción de añadir una nueva etiqueta al recurso. 6. El usuario escoge, de entre su lista de etiquetas, la etiqueta que desea añadir al recurso. 7. La etiqueta se asocia al recurso seleccionado.

Caso de uso	Visualizar lista de recursos de una etiqueta	Identificador	CU-M-8
Actores	Usuario registrado		
Escenario	El usuario desea etiquetar un recurso con una etiqueta de su cuenta.		
Precondiciones	<ul style="list-style-type: none"> – El usuario debe tener una cuenta en la aplicación web ColDes. – El usuario debe haber añadido a su cuenta algún recurso. – El usuario debe haber añadido a su cuenta alguna etiqueta. – El usuario debe haber etiquetado algún recurso de su cuenta. 		
Curso básico	<ol style="list-style-type: none"> 1. El usuario accede a la zona de etiquetas de la aplicación. 2. El usuario visualiza la lista de etiquetas asociados a su cuenta. 3. El usuario selecciona la etiqueta de la que desea obtener la lista de recursos asociados. 4. Se muestra la lista de recursos asociados a la etiqueta. 		

Caso de uso	Eliminar etiquetas	Identificador	CU-M-9
Actores	Usuario registrado		
Escenario	El usuario desea eliminar un recurso de su cuenta.		
Precondiciones	<ul style="list-style-type: none"> – El usuario debe tener una cuenta en la aplicación web ColDes. – El usuario debe haber añadido a su cuenta alguna etiqueta. 		
Curso básico	<ol style="list-style-type: none"> 1. El usuario accede a la zona de etiquetas de la aplicación. 2. El usuario visualiza la lista de etiquetas asociados a su cuenta. 3. El usuario selecciona la etiqueta que desea eliminar. 4. El usuario selecciona la opción de eliminar la etiqueta seleccionada. 5. La etiqueta se elimina de la cuenta del usuario. 		

A continuación, se muestra las matrices de trazabilidad que relaciona cada uno de los requisitos presentados en el apartado *Definición de requisitos* con los casos de uso que se han presentado en el presente apartado. En la *Tabla 12* se presenta la matriz para el caso de la aplicación web, mientras que en la *Tabla 13* se presenta la matriz para el caso de la aplicación móvil. Cabe destacar que sólo se han tenido en cuenta los requisitos funcionales, pues son los que pueden ser relacionados con los casos de uso por ser los que definen la funcionalidad del sistema.

Como se puede observar en ambas matrices, todos los requisitos se encuentran relacionados con un caso de uso. Los casos de uso, a su vez, pueden relacionarse con más de un requisito.

	CU-W-1	CU-W-2	CU-W-3	CU-W-4	CU-W-5	CU-W-6	CU-W-7	CU-W-8	CU-W-9	CU-W-10	CU-W-11
RF-W-1	X										
RF-W-2	X										
RF-W-3	X										
RF-W-4	X										
RF-W-5		X									
RF-W-6		X									
RF-W-7		X	X								
RF-W-8			X								
RF-W-9			X								
RF-W-10			X								
RF-W-11				X							
RF-W-12							X				
RF-W-13								X			
RF-W-14							X				
RF-W-15									X		
RF-W-16										X	
RF-W-17											X
RF-W-18					X						
RF-W-19					X						
RF-W-20						X					

Tabla 12: Matriz de trazabilidad: Requisitos funcionales vs. Casos de uso. Aplicación web.

	CU-M-1	CU-M-2	CU-M-3	CU-M-4	CU-M-5	CU-M-6	CU-M-7	CU-M-8	CU-M-9
RF-M-1									
RF-M-2									
RF-M-3									
RF-M-4									
RF-M-5									
RF-M-6									
RF-M-7									
RF-M-8									
RF-M-9									
RF-M-10									
RF-M-11									
RF-M-12									
RF-M-13									
RF-M-14									
RF-M-15									
RF-M-16									

Tabla 13: Matriz de trazabilidad: Requisitos funcionales vs. Casos de uso. Aplicación móvil.

Diseño

La fase de diseño tiene por objetivo anticipar y guiar el proceso de producción del sistema, creando modelos o representaciones de las entidades que se construyen posteriormente. A partir de la fase de diseño se busca la implementación de un sistema que cumpla los requisitos que se han establecido en la fase de análisis, y que se ajuste a las condiciones dadas por las restricciones operativas y tecnológicas.

En los siguientes puntos se define la arquitectura escogida para el sistema, así como las tecnologías utilizadas en el desarrollo del mismo. Más adelante, se establece el diseño detallado del sistema, en el que se describe el diseño de las capas de persistencia, modelo e interfaz.

Diseño de sistema

A continuación, se presenta la arquitectura del sistema y las tecnologías utilizadas en el desarrollo del mismo. Tal y como se establece en el apartado 4.1, el sistema desarrollado cuenta con dos aplicaciones: una aplicación web y una aplicación móvil.

El entorno desarrollado se basa en la arquitectura cliente-servidor. Se trata de un modelo de aplicación distribuida en el que las tareas se reparten entre el servidor y el cliente. El servidor es el encargado de proveer los recursos y servicios que son demandados por el cliente.

La arquitectura de la aplicación web ya se definió en la memoria del proyecto fin de carrera de ColDes [1], por lo que las siguientes líneas se centrarán en definir los cambios que se han realizado respecto a la arquitectura original, así como en establecer el esquema de comunicación entre la aplicación web y la aplicación móvil.

Por otra parte, la aplicación móvil se ha desarrollado en tecnología Android. Tal como se presenta en la sección *Revisión de tecnologías móviles*, se trata de un sistema operativo diseñado para *smartphones* y *tablets* cuyas aplicaciones se desarrollan en lenguaje de programación Java. En cuanto a la versión utilizada para el desarrollo de la aplicación móvil, se ha utilizado la versión *IceCream Sandwich*, o 4.0, ya que, desde su aparición, la cuota de mercado de esta versión ha aumentado hasta situarse en el 27.5% de los dispositivos con Android (véase *Figura 12*). Por otro lado, se trata de uno de los requisitos no funcionales que debe cumplir la aplicación móvil.

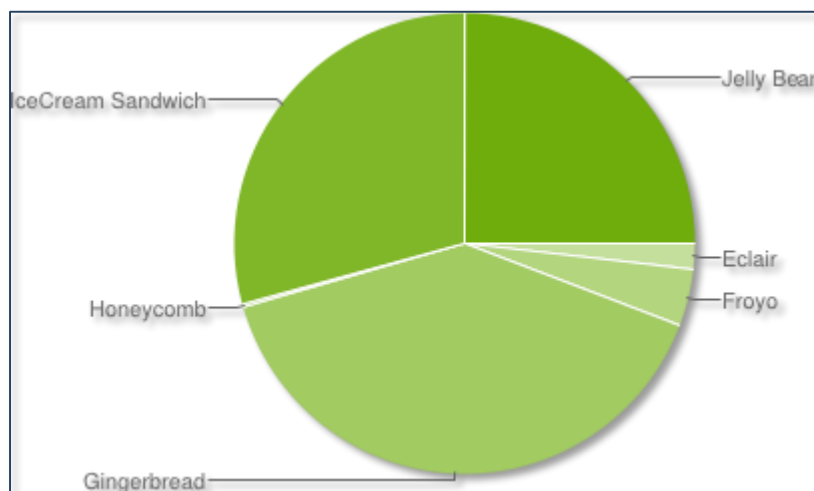


Figura 12: Distribución de la cuota de mercado entre las diferentes versiones de Android⁹.

Los cambios que se han realizado en la arquitectura del sistema con respecto a la aplicación ColDes original son los siguientes:

- Distribuir la aplicación para independizar el modelo de negocio de la capa de persistencia.
- Añadir una capa de comunicación intermedia que permite la comunicación con la capa de persistencia a través de diferentes tecnologías.

En cuanto al primero de los cambios, se ha separado la capa de persistencia de la propia aplicación, tal como se muestra en la *Figura 13*. A su vez, esta capa de persistencia se ha dividido en dos partes: una base de datos relacional y un sistema de ficheros. La primera conserva la misma

⁹ Fuente: http://commons.wikimedia.org/wiki/File:Android_chart.png

función que tenía anteriormente en ColDes, permitiendo almacenar toda la información relacionada con los usuarios, los *workspaces* y los diseños. Por otra parte, se ha incluido el sistema de ficheros, que permite almacenar físicamente los recursos en forma de ficheros. Por tanto, se mantienen distintos elementos de persistencia en un único sistema.

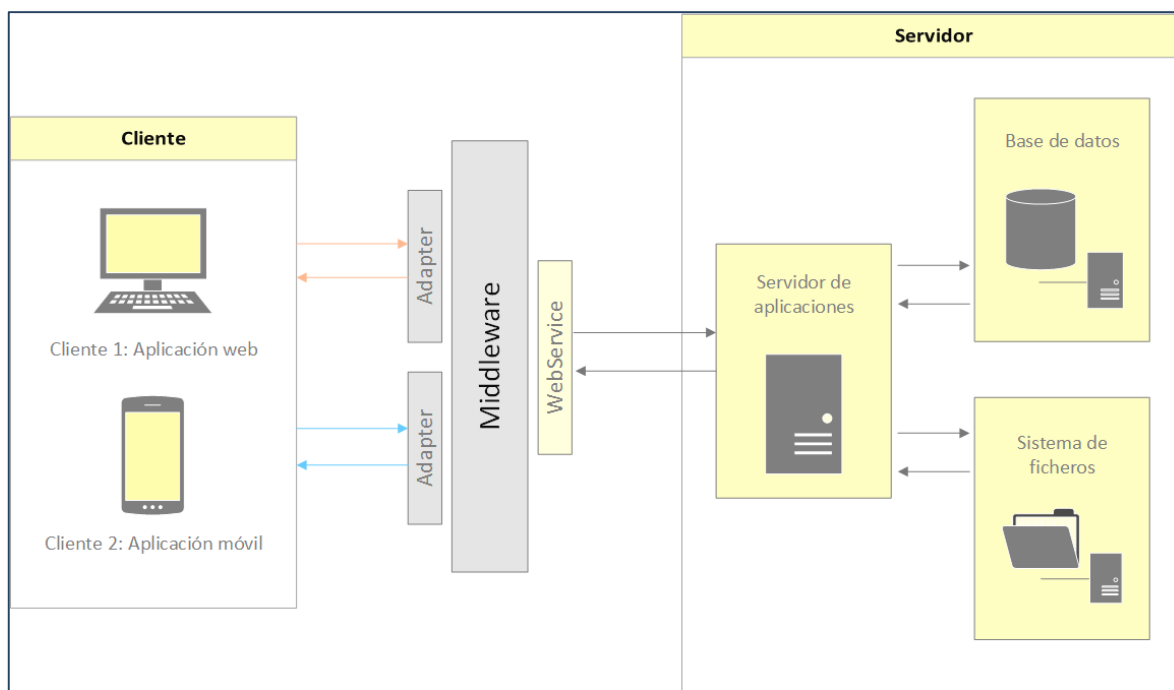


Figura 13: Arquitectura del entorno desarrollado.

Esta diferenciación permite disminuir considerablemente los tiempos asociados a la subida y bajada de recursos a través de la aplicación móvil. En la aplicación original, los diseños se guardaban completamente en la base de datos en forma de *array* de bytes. Sin embargo, se trata de una forma muy ineficiente de guardar información que puede ser de gran tamaño. Por este motivo, el almacenamiento de los recursos se realiza actualmente en el sistema de ficheros que se encuentra alojado en el servidor. Por otra parte, sólo es necesario almacenar en la base de datos la dirección en la que se encuentra el recurso. Esta transformación permite que los recursos puedan ser guardados y descargados de manera más rápida a través del dispositivo móvil, ya que no es necesario serializar el recurso antes de leerlo o escribirlo.

El segundo de los cambios ha consistido en introducir una nueva capa intermedia de abstracción que permita la comunicación entre clientes y entre éstos y la capa de persistencia. Esta capa se denomina *middleware*. Se trata de un *framework* software basado en el patrón de diseño *adapter* [17, p. 139]. Este patrón estructural es utilizado para convertir la interfaz de una clase en la interfaz correspondiente a un cliente concreto. En este caso, tal como se muestra en la [Figura 13](#), existe un *adapter* específico para cada tipo de cliente, que permite que la conexión con

el *Web service* se pueda realizar por medio de diferentes tecnologías, sin necesidad de tener que duplicar la capa de comunicación con la capa de persistencia. En concreto, permite que la comunicación se pueda realizar a partir de la tecnología utilizada en la parte servidor de la aplicación web (Java), y a partir de la tecnología utilizada en la aplicación móvil (Android).

El *Web service*, por su parte, permite que distintas aplicaciones software desarrolladas en lenguajes de programación distintas, y ejecutadas en diferentes plataformas, puedan intercambiar datos y acceder a un sistema de persistencia común. Existen numerosos protocolos utilizados por los *Web services* [18]. Para el sistema desarrollado en el presente proyecto se han utilizado dos de ellos: SOAP [19] y REST [20] bajo el lenguaje de programación Java.

El primero de ellos, SOAP (*Simple Object Access Protocol*), se basa en el intercambio de datos en formato XML entre diferentes procesos. Para conseguir ese propósito, es necesario que los mensajes SOAP cuenten con una estructura concreta. Concretamente, debe incluir el elemento base del mensaje, denominado *Envelope*, que contenga los elementos *Header* y *Body*. Estos últimos contienen la información del mensaje y la carga de datos del mensaje, respectivamente.

El segundo de los protocolos, REST (*Representational State Transfer*) se considera como un estilo de arquitectura para el diseño de aplicaciones *Web*. Está basado en la idea de simplificar los protocolos más complejos, como SOAP, utilizando HTTP para realizar la comunicación entre diferentes procesos.

El protocolo SOAP se ha utilizado en el *Web service* en todos aquellos métodos que permiten al cliente guardar u obtener datos de la base de datos. Por otro lado, el protocolo REST se ha utilizado para los métodos que permiten guardar y obtener recursos del sistema de ficheros, ya que este proceso se ve notablemente acelerado al no ser necesario serializar el contenido de un objeto, como ocurre con SOAP.

Un ejemplo del funcionamiento de la arquitectura completa del sistema se muestra en la [Figura 14](#) en forma de diagrama de secuencia. En concreto, se describe la secuencia realizada en la gestión de recursos del sistema, tanto para la aplicación web como para la aplicación móvil.

El proceso se inicia cuando el usuario quiere añadir un recurso a su cuenta mediante la aplicación móvil. En ese momento, el *adapter* de la aplicación móvil obtiene la interfaz apropiada a este tipo de aplicación (en tecnología Android) y llama al método *Web service* que permite guardar un nuevo recurso en el sistema de persistencia. El primer paso es guardar el recurso en forma de fichero utilizando el protocolo REST de comunicación. Este proceso devuelve la dirección del recurso en el sistema de ficheros. A continuación, se guarda la información del recurso, incluida la dirección en la que ha sido guardado, en la base de datos, mediante el protocolo SOAP. Si la

operación se ha realizado correctamente, se muestra al usuario un mensaje de confirmación de la operación.

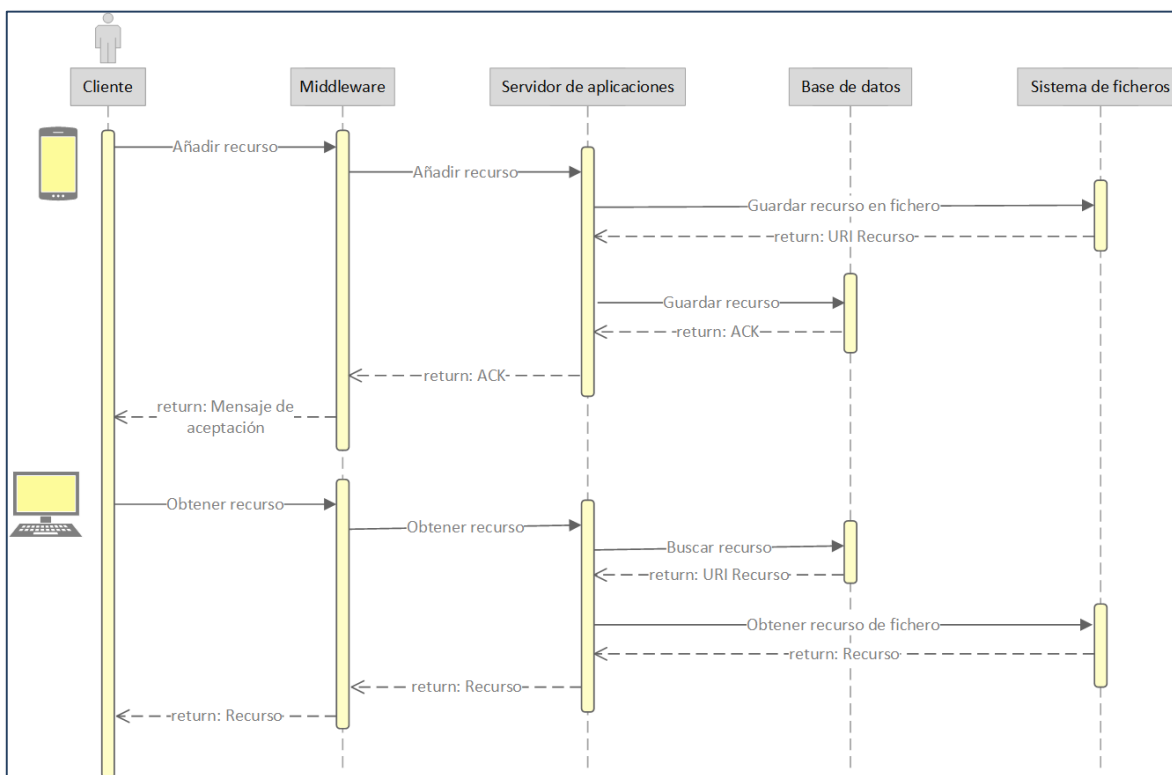


Figura 14: Diagrama de secuencia: administración de recursos.

El proceso es similar cuando el usuario quiere obtener un recurso, ya sea a través de la aplicación web o a través de la aplicación móvil. Si se realiza con la aplicación web, el *adapter* correspondiente obtiene la interfaz adecuada (en lenguaje Java) y llama al método del *Web service* que permite descargar un recurso que se encuentre en el sistema de persistencia del sistema. Éste se encarga de buscar el recurso en la base de datos y obtener la dirección en la que se encuentra el recurso físico en el sistema de ficheros, mediante el protocolo SOAP. A continuación, mediante el protocolo REST, se obtiene el recurso de la dirección adquirida de la base de datos y se envía al cliente.

Diseño detallado

En el siguiente apartado se muestra el diseño detallado del sistema, describiendo el diseño de las capas de persistencia e interfaz.

Capa de persistencia

Como se ha descrito en el apartado de *Diseño de sistema*, se han realizado cambios en la capa de persistencia con respecto a la aplicación ColDes original. Este apartado tiene por objetivo definir y argumentar esos cambios.

1. Cambios estructurales en la capa de persistencia

El primero de los cambios es la división de la capa de persistencia en dos partes diferenciadas: una base de datos relacional y un sistema de ficheros. El sistema de ficheros no es más que un espacio del servidor dedicado al almacenamiento de los recursos de la aplicación. La estructura de directorios del sistema de ficheros se muestra en la *Figura 15*. Como se puede observar, existe un directorio destinado a cada uno de los tipos de recurso que es posible añadir a la aplicación. Así, los recursos de vídeo se encontrarán en el directorio “video”, mientras que los de sonido se encontrarán en “sound”.

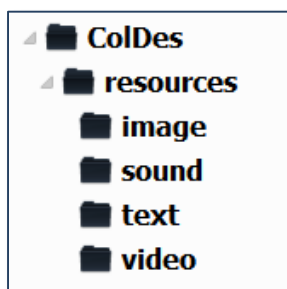


Figura 15: Estructura de directorios del sistema de ficheros.

2. Cambios en la base de datos

En cuanto a la base de datos, se han realizado varias modificaciones con respecto a la original. En la *Figura 16* se muestra el diagrama físico de datos de la nueva base de datos.

Como se puede observar, la mayor parte de la base de datos se mantiene con respecto a la original. La parte que aparece destacada en el diagrama se corresponde con la parte añadida a la base de datos original. Se trata de las tablas necesarias para la gestión de los recursos de la base de datos, así como las tablas necesarias para mantener los diagramas de afinidad creados por el usuario en el sistema de persistencia.

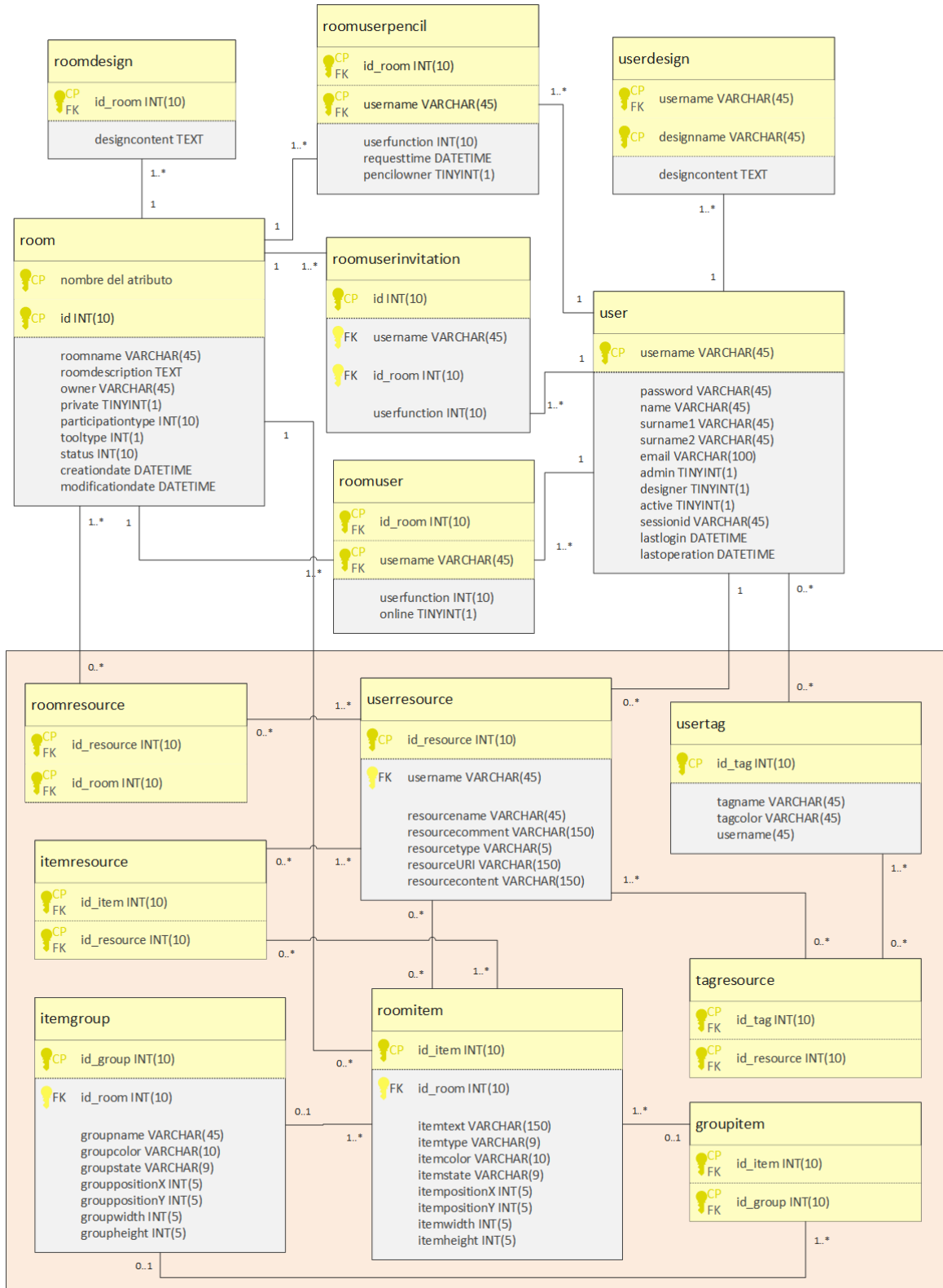


Figura 16: Diagrama físico de datos.

A continuación, se describe la función de cada una de las tablas añadidas a la base de datos, así como los campos de los que está formada.

- **Tabla *userresource*:** Se trata de la tabla en la que se almacenan los datos referentes a los recursos añadidos por el usuario a través de la aplicación móvil. La clave primaria se corresponde con un identificador de recurso.

Tabla <i>userresource</i>		
Campo	Tipo	Descripción
id_resource	INT(10)	Identificador unívoco del recurso. Se trata de un campo que se autoincrementa al realizar una inserción en la tabla.
userresource	VARCHAR(45)	Nombre de usuario del usuario que ha añadido el recurso. Se corresponde con el campo <i>username</i> de la tabla <i>user</i> .
resourcenname	VARCHAR(45)	Nombre del recurso. No es necesario que sea único.
resourcecomment	VARCHAR(150)	Comentario sobre el recurso añadido por el usuario en la inserción del recurso en la tabla.
resourcetype	VARCHAR(5)	Tipo de recurso. Puede tomar los valores IMAGE, SOUND, VIDEO o TEXT.
resourceURI	VARCHAR(150)	Dirección del recurso en el sistema de ficheros. Tomará valor para los recursos de imagen, sonido y vídeo.
resourcecontent	VARCHAR(150)	Contenido del recurso. Tendrá valor únicamente para los recursos de texto.

Tabla 14: Descripción de la tabla *userresource* de la base de datos.

- **Tabla *roomresource*:** Esta tabla representa la relación entre los recursos y los *workspaces*. Dado que puede haber más de un recurso asociado a un *workspace*, y un recurso puede estar asociado a varios *workspaces*, es necesario crear una tabla intermedia.

Tabla <i>roomresource</i>		
Campo	Tipo	Descripción
id_resource	INT(10)	Identificador unívoco del recurso. Se corresponde con el campo <i>id_resource</i> de la tabla <i>userresource</i> .
id_room	INT(10)	Identificador unívoco del <i>workspace</i> . Se corresponde con el campo <i>id_room</i> de la tabla <i>room</i> .

Tabla 15: Descripción de la tabla *roomresource* de la base de datos.

- **Tabla *usertag*:** En esta tabla se almacenan los datos referentes a los etiquetas creadas por el usuario a través de la aplicación móvil. La clave primaria se corresponde con un identificador único de la etiqueta.

Tabla <i>usertag</i>		
Campo	Tipo	Descripción
id_tag	INT(10)	Identificador unívoco de la etiqueta. Se trata de un campo que se autoincrementa al realizar una inserción en la tabla.
tagname	VARCHAR(45)	Nombre de la etiqueta. No es necesario que sea único, pues la etiqueta se identifica mediante el campo anterior.
tagcolor	VARCHAR(45)	Color de la etiqueta en notación hexadecimal.
username	VARCHAR(45)	Nombre de usuario del usuario que ha creado la etiqueta. Se corresponde con el campo <i>username</i> de la tabla <i>user</i> .

Tabla 16: Descripción de la tabla *usertag* de la base de datos.

- **Tabla *tagresource*:** Esta tabla representa la relación entre los recursos y las etiquetas. Dado que puede haber más de un recurso etiquetado por una etiqueta, y un recurso puede estar etiquetado por varias etiquetas, es necesario crear una tabla intermedia.

Tabla <i>tagresource</i>		
Campo	Tipo	Descripción
id_resource	INT(10)	Identificador unívoco del recurso. Se corresponde con el campo <i>id_resource</i> de la tabla <i>userresource</i> .
id_tag	INT(10)	Identificador unívoco de la etiqueta. Se corresponde con el campo <i>id_tag</i> de la tabla <i>usertag</i> .

Tabla 17: Descripción de la tabla *tagresource* de la base de datos.

- **Tabla *roomitem*:** En esta tabla se almacenan los datos referentes a los ítem de información creados por el usuario en la construcción de un diagrama de afinidad. La clave primaria se corresponde con un identificador único de la etiqueta.

Tabla <i>roomitem</i>		
Campo	Tipo	Descripción
id_item	INT(10)	Identificador unívoco del ítem de información. Se trata de un campo que se autoincrementa al realizar una inserción en la tabla.
id_room	INT(10)	Identificador del <i>workspace</i> en el que se ha creado el ítem. Se corresponde con el campo <i>id_room</i> de la tabla <i>room</i> .
itemtext	VARCHAR(150)	Texto asociado al ítem.
itemtype	VARCHAR(9)	Tipo de ítem de información. Puede tomar los valores CONTAINER o BASE, según tenga asociado algún recurso o no, respectivamente.
itemcolor	VARCHAR(10)	Color asociado al ítem de información.
itemstate	VARCHAR(9)	Estado del ítem de información en el diagrama de afinidad. Puede tomar los valores NORMAL, MINIMIZED, COLLAPED o GROUPED, según el ítem esté en su estado normal, minimizado, colapsado o agrupado, respectivamente.
itempositionX	INT(5)	Posición del ítem en el eje X del <i>workspace</i> .
itempositionY	INT(5)	Posición del ítem en el eje Y del <i>workspace</i> .
itemwidth	INT(5)	Anchura del ítem de información.
itemheight	INT(5)	Altura del ítem de información.

Tabla 18: Descripción de la tabla *roomitem* de la base de datos.

- **Tabla *itemgroup*:** Se trata de la tabla en la que se almacenan los datos referentes a los grupos de ítems creados por el usuario en la construcción de un diagrama de afinidad. La clave primaria se corresponde con un identificador único del grupo.

Tabla <i>itemgroup</i>		
Campo	Tipo	Descripción
id_group	INT(10)	Identificador unívoco del grupo de ítems. Se trata de un campo que se autoincrementa al realizar una inserción en la tabla.
id_room	INT(10)	Identificador del <i>workspace</i> en el que se ha creado el grupo de ítems. Se corresponde con el campo <i>id_room</i> de la tabla <i>room</i> .
groupname	VARCHAR(45)	Nombre del grupo de ítems. No es necesario que sea único, pues el grupo se identifica mediante el campo <i>id_group</i> .
groupcolor	VARCHAR(10)	Color asociado al grupo.
groupstate	VARCHAR(9)	Estado del grupo de ítems en el diagrama de afinidad. Puede tomar los valores NORMAL, MINIMIZED, o COLLAPED, según el grupo se encuentre en su estado normal, minimizado, o colapsado respectivamente.
grouppositionX	INT(5)	Posición del grupo en el eje X del <i>workspace</i> .
grouppositionY	INT(5)	Posición del grupo en el eje Y del <i>workspace</i> .
groupwidth	INT(5)	Anchura del grupo de información.
groupheight	INT(5)	Altura del grupo de información.

Tabla 19: Descripción de la tabla *itemgroup* de la base de datos.

- **Tabla *groupitem*:** Esta tabla representa la relación entre los ítems de información y los grupos de ítems. Un ítem puede pertenecer a un grupo o no pertenecer a ninguno. Por otra parte, un grupo de ítems contiene uno o varios ítems. Teniendo en cuenta esto, no sería necesario crear una tabla intermedia para asociar ítems de información con los grupos a los que pertenecen. Sin embargo, inicialmente un ítem no tiene por qué estar agrupado, por lo que una clave ajena para indicar el grupo al que pertenece un ítem en la tabla *roomitem* no podría ser utilizada, ya que inicialmente no tendría valor. Por ello, es necesario crear una tabla intermedia para representar la relación entre los ítems y los grupos de ítems.

Tabla <i>groupitem</i>		
Campo	Tipo	Descripción
id_item	INT(10)	Identificador unívoco del ítem de información. Se corresponde con el campo <i>id_item</i> de la tabla <i>roomitem</i> .
id_group	INT(10)	Identificador unívoco del grupo de ítems. Se corresponde con el campo <i>id_group</i> de la tabla <i>itemgroup</i> .

Tabla 20: Descripción de la tabla *groupitem* de la base de datos.

- **Tabla *itemresource*:** Esta tabla representa la relación entre los recursos y los ítems de información. Dado que un ítem puede tener asociado más de un recurso, y un recurso puede estar asociado a varios ítems, es necesario crear una tabla intermedia.

Tabla <i>itemresource</i>		
Campo	Tipo	Descripción
id_item	INT(10)	Identificador unívoco del ítem de información. Se corresponde con el campo <i>id_item</i> de la tabla <i>roomitem</i> .
id_resource	INT(10)	Identificador unívoco del recurso. Se corresponde con el campo <i>id_resource</i> de la tabla <i>userresource</i> .

Tabla 21: Descripción de la tabla *itemresource* de la base de datos.

3. Distribución de la capa de persistencia

Otro de los cambios realizados en la capa de persistencia del sistema es la distribución de la misma, con el fin de separar el modelo de negocio de la capa de persistencia. Como se describe en el apartado de *Diseño de sistema*, la comunicación entre el cliente y la capa de persistencia se realiza a partir del *middleware*. A continuación, se presentan los patrones de diseño utilizados para facilitar el acceso al sistema de persistencia desde el cliente a través del *middleware*.

Patrón de diseño *Abstract Factory*

Abstract Factory es un patrón de diseño creacional. Este tipo de patrones permiten resolver problemas sobre cómo crear instancias de las clases del sistema. En concreto, el patrón *Abstract Factory* “proporciona una interfaz para crear familias de objetos relacionados o dependientes sin especificar sus clases concretas” [17, p. 87].

El patrón *Abstract Factory* cuenta con una estructura en la que existen los siguientes participantes:

- ***AbstractFactory***: Define la interfaz de los objetos de tipo *ConcreteFactory*. Se trata de una interfaz en la que se definen las operaciones en las que se crean los productos abstractos.
- ***ConcreteFactory***: Se trata de las diferentes familias de productos. Deben contener las operaciones para crear los productos concretos, tal como se define en el *AbstractFactory*.
- ***AbstractProduct***: Define la interfaz de los objetos de tipo *ConcreteProduct*. Se trata de una interfaz en la que se definen las operaciones que deben completar los productos concretos.
- ***ConcreteProduct***: Se trata de la implementación de los productos creados por el *ConcreteFactory* correspondiente. Implementa la interfaz *AbstractProduct*.
- ***Client***: Clase que llama a la interfaz *AbstractFactory* para obtener instancias de los productos concretos adecuados.

Antes de explicar cómo se ha utilizado el patrón *Abstract Factory* en la arquitectura del sistema, es necesario explicar los patrones de diseño DAO y DTO.

Patrón de diseño DAO

DAO, o *Data Access Object*, es un patrón de diseño que permite “abstraer y encapsular todos los accesos al almacenamiento persistente. El *Data Access Object* gestiona la conexión con la fuente de datos para obtener y almacenar los datos” [21, p. 462]. Por tanto, pueden ser utilizados por cualquier tipo de cliente sin que éste conozca el destino final de los datos.

Patrón de diseño DTO

DTO, o *Data Transfer Object*, es un patrón de diseño que “permite transferir datos entre subsistemas de una aplicación software” [22]. Los DTOs son objetos simples del contexto del sistema que no contienen ningún tipo de comportamiento. Únicamente definen los atributos del objeto y los métodos que permiten acceder a éstos o modificarlos: los *getters* y *setters*.

Utilización de los patrones *Abstract Factory*, DAO y DTO en la arquitectura del sistema

Tras definir los patrones de diseño utilizados en la arquitectura del sistema, se va a describir la manera en que se han utilizado, así como las ventajas que supone la utilización de cada uno de ellos.

Para establecer cómo se han utilizado los patrones definidos anteriormente se ha creado un diagrama de clases parcial que presenta algunas de las clases creadas para la comunicación entre el cliente y el sistema de persistencia. Este diagrama se muestra en la [Figura 17](#).

En el presente proyecto, se ha utilizado el patrón *Abstract Factory* junto con el patrón DAO con el objetivo de permitir que el sistema de persistencia pueda estar basado en cualquier estructura de datos persistente. Así, en el diagrama de la [Figura 17](#), se muestra un ejemplo para el caso en el que la persistencia del sistema estuviese basada en ficheros XML o en una base de datos relacional MySQL.

Como se puede observar en el diagrama, se muestra la relación entre los participantes del patrón *Abstract Factory* y las clases del sistema. De esta manera, se cuenta con una clase *ADAOFactory* que se corresponde con el participante *AbstractFactory*. En esta clase se define la interfaz que deben cumplir los *ConcreteFactory* que la implementen, es decir, las clases que van a establecer qué tipo de sistema de persistencia se utiliza. Por ejemplo, en el diagrama se establecen las clases *MySQLDAOFactory* y *XMLDAOFactory* como clases del tipo *ConcreteFactory*. Esto quiere decir que el entorno permitiría que el sistema de consistencia estuviese basado en XML o en MySQL.

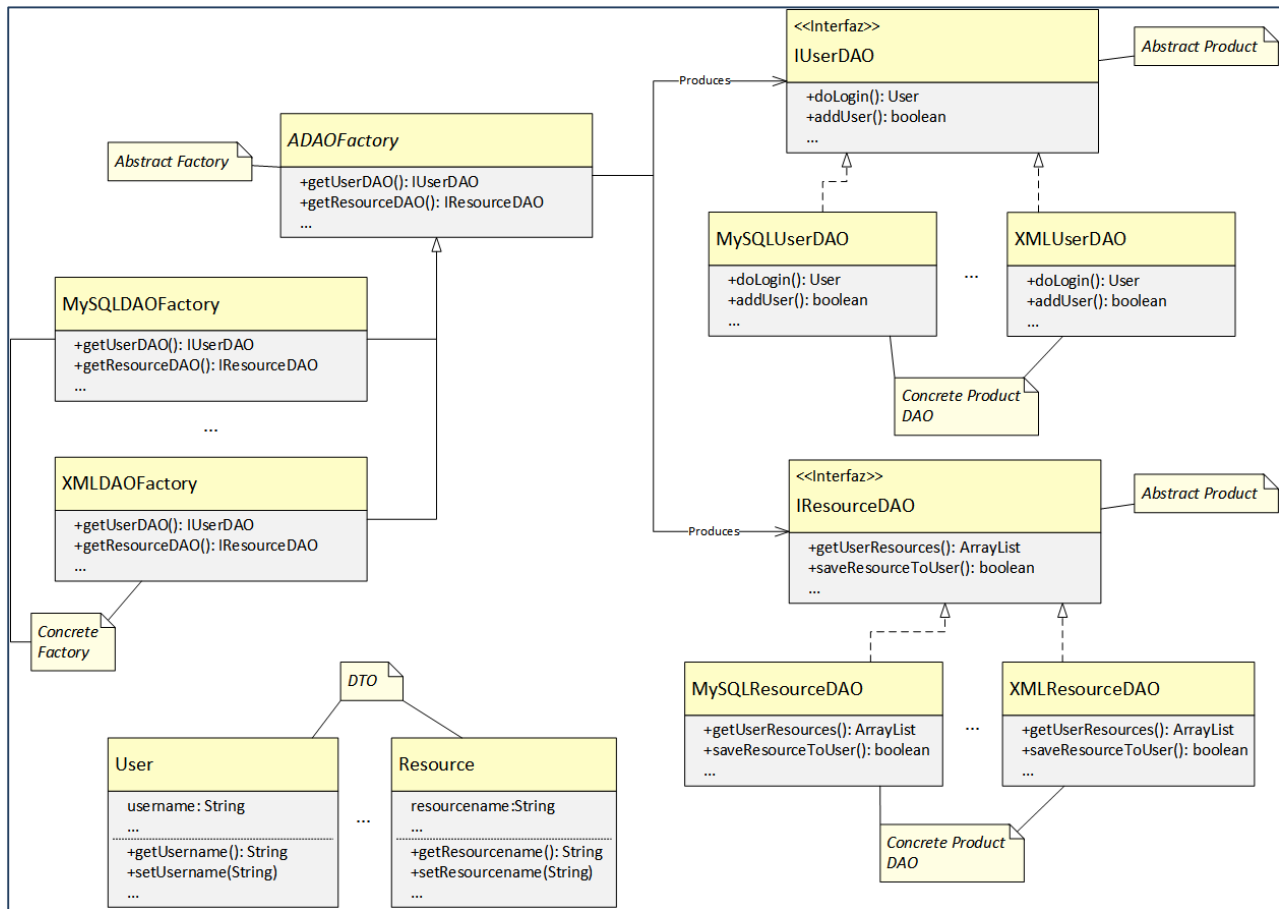


Figura 17: Diagrama de clases: Utilización de patrones de diseño.

Los *AbstractProduct* se corresponden con las interfaces de los DAO. En este proyecto, existe un DAO por cada DTO. Por su parte, existe un DTO por cada objeto del dominio que es necesario mantener en el sistema de persistencia. Por tanto, existe un DTO y un DAO para cada uno de los siguientes objetos:

- Usuario.
- Workspace.
- Recurso.
- Diseño.
- Píxel.
- Ítem de información.
- Grupo de ítems.
- Etiqueta.

Los DTO del sistema contienen los atributos de los objetos del dominio. Estos atributos no son más que las propiedades de la entidad, tal como se define en el diagrama entidad-relación de la [Figura 16](#). Por tanto, en el DTO que describe a un usuario es necesario que existan atributos para el nombre de usuario, la contraseña, el nombre, los apellidos, etc. De igual manera, el DTO contiene los métodos para obtener y modificar esos atributos. En el diagrama de la [Figura 17](#) se muestra el ejemplo para los DTO que describen a un usuario y a un recurso.

Como se ha comentado líneas arriba, existe un DAO asociado a cada DTO. Estos DAO contienen las operaciones que se pueden realizar sobre el sistema de persistencia. En este caso, se trata de las operaciones de creación, modificación y eliminación de objetos del dominio en la base de datos MySQL. En el diagrama de la [Figura 17](#) se muestra el ejemplo de la interfaz de los DAO de usuario y recurso, así como los DAO asociados a un sistema de persistencia en MySQL y en XML (los *ConcreteProduct* asociados al patrón *Abstract Factory*).

En cuanto a la comunicación entre el cliente y el sistema de persistencia, para el cliente toda la arquitectura basada en patrones de diseño es transparente. Únicamente instancia el *ConcreteFactory* asociado al sistema de persistencia adecuado y realiza las llamadas al DAO del objeto que desea añadir, modificar o eliminar.

Tras describir el uso de los patrones de diseño en la arquitectura del sistema, se van a presentar las ventajas que esta arquitectura presenta.

Una de las ventajas de utilizar el patrón *Abstract Factory* es que para añadir un nuevo tipo de sistema de persistencia sólo sería necesario crear una nueva clase *ConcreteFactory* y las clases *ConcreteProduct* necesarias. En este caso, al utilizar el patrón DAO, si se quisiese añadir un sistema de persistencia basado en ficheros XML, por ejemplo, sólo habría que añadir la clase *XMLDAOFactory* y las clases asociadas a los DAO, como se muestra en el diagrama de la [Figura 17](#).

En cuanto al uso del patrón DAO, su principal ventaja es que, al encapsular los objetos de acceso a la base de datos, se podría cambiar en cualquier momento la tecnología de persistencia sin necesidad de modificar estos objetos. Por otra parte, se aísla la conexión con el sistema de persistencia a una capa diferenciada del resto de la lógica de negocio.

Con respecto al patrón DTO, la principal ventaja de su uso es que permite aumentar el encapsulamiento de la capa de negocio. En el caso concreto de este proyecto, los DTO ayudan en la comunicación del cliente con el sistema de persistencia, sobre todo en la aplicación móvil. Cuando la parte cliente hace una llamada al *Web service* utilizando el protocolo SOAP, es necesario convertir los objetos Java a su representación JSON. El recibir la llamada, el *Web service* debe realizar la conversión inversa. Dado que se cuenta con los DTO, es muy sencillo *parsear* el JSON al objeto adecuado.

Interfaz

El desarrollo de la interfaz, tanto de la aplicación web como de la aplicación móvil, se ha realizado a través de la implementación de distintos prototipos. En este apartado se muestra la evolución de los prototipos desarrollados hasta llegar a la interfaz final del sistema.

Aplicación web

En el caso de la aplicación web, dado que se partía de la interfaz de la aplicación ColDes, sólo ha sido necesario realizar el prototipo de la interfaz del *workspace* para la creación de diagramas de afinidad. En la [Figura 18](#) se muestra la barra de herramientas que se presentó en el primer prototipo del *workspace* de creación de diagramas de afinidad. Se trata de un *ribbon* [23], un tipo de interfaz de usuario formado por una cinta en la parte superior de una ventana donde se recogen todas las funciones que se pueden realizar en esa ventana. De esta manera, se aumentaría el espacio de trabajo disponible en el *workspace* con respecto a las antiguas salas.

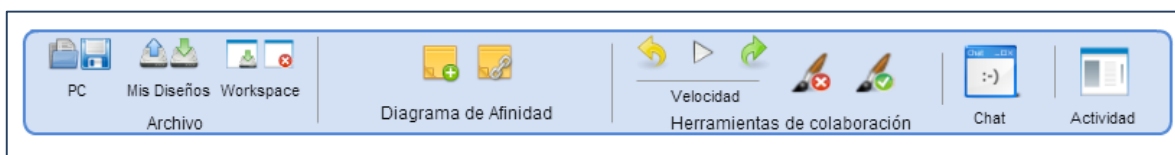


Figura 18: Prototipo del *workspace* de diagrama de afinidad. Primera iteración (izquierda).

Como se puede observar, todas las funciones para la creación de diagramas de afinidad se presentan en el *ribbon*. Por otra parte, se decidió ocultar el chat y el cuadro de actividad del *workspace*, de manera que sólo fuesen visibles al pulsar sobre el icono correspondiente. En ese caso, se añadiría un nuevo cuadro flotante que el usuario podría disponer en cualquier parte del *workspace*, sin limitar el espacio de trabajo.

En la [Figura 19](#) se presenta la interfaz final del *workspace* para la creación de diagrama de afinidad. Se trata de la segunda iteración de prototipado. Como se puede observar, no se han realizado cambios sustanciales entre las dos iteraciones de prototipado. Sin embargo, si se presentan pequeños cambios relacionados con la iconografía utilizada.

El primero de ellos fue la sustitución del icono para la agrupación de ítems, ya que el que se utilizó en el primer prototipo sugería el enlace de ítems, y no su agrupación. Por otro lado, se cambiaron los iconos del chat y la actividad del *workspace* por otros más significativos.

Por último, se decidió aumentar aún más el tamaño del espacio de trabajo. Para ello, se modificó el *ribbon* para que fuera colapsable. De esta forma, el único espacio que no se puede utilizar para la creación de diagramas de afinidad es el ocupado por el sistema de pestañas.

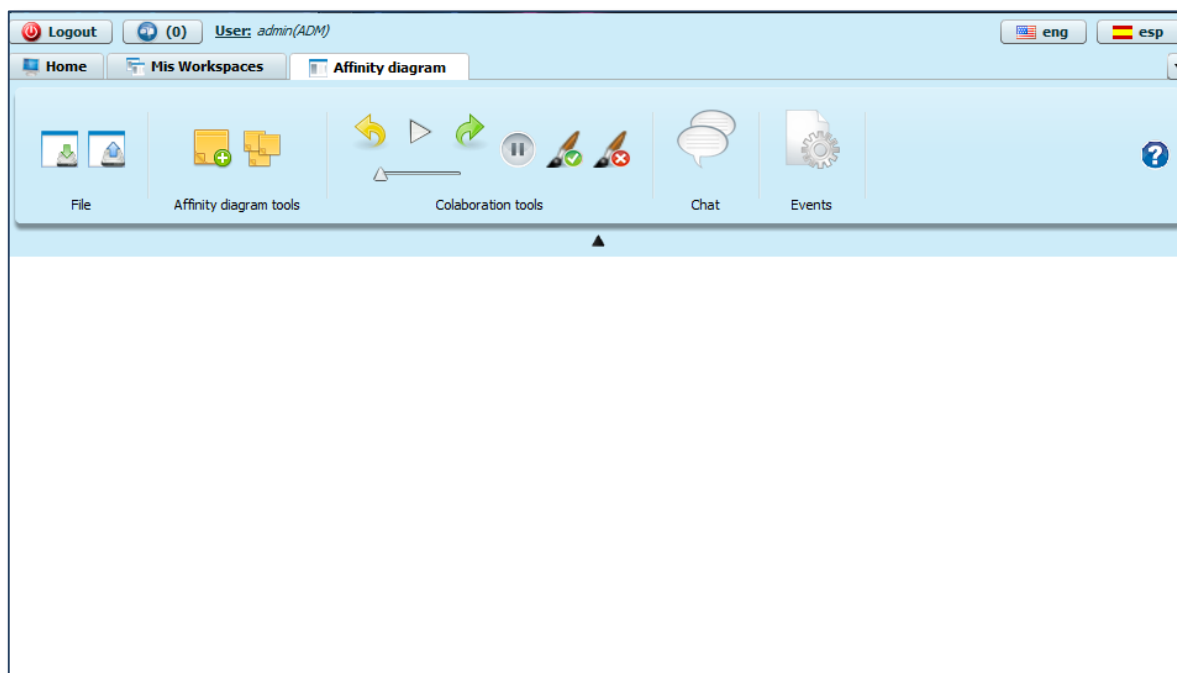


Figura 19: Interfaz del *workspace* de diagrama de afinidad. Aplicación final.

Aplicación móvil

En el apartado dedicado a explicar el modelo de proceso seguido en el presente proyecto se adelantó que se han realizado tres iteraciones en el proceso de prototipado de la aplicación móvil. Las dos primeras no llegaron al proceso de implementación, mientras que el último prototipado se corresponde con la versión final de la aplicación.

En el presente apartado se describen algunos de los cambios realizados entre las distintas iteraciones del proceso de prototipado. El prototipo final se puede consultar en el capítulo [Anexo IV. Prototipo](#).

En la [Figura 20](#) se muestra la comparación de la pantalla principal de la aplicación móvil para las dos primeras iteraciones del prototipo. Como se puede observar, en la primera iteración, la importancia de la aplicación radicaba en la gestión de las salas y los diseños, mientras que en la segunda iteración se consideró que la importancia de la aplicación residía en la gestión de recursos. Por otra parte, se decidió que el término *workspace* describía de manera más precisa el propósito del espacio de trabajo común que *room*. Estos *workspace* quedaron en segundo plano al incluirlos en la pestaña de la derecha. De este modo, la aplicación muestra por defecto la información de los recursos. Por otro lado, se eliminó la información de los diseños de la aplicación, pues se consideró que no era relevante para la aplicación móvil.



Figura 20: Prototipo de la pantalla principal. Primera iteración (izquierda) y segunda iteración (derecha).

Otro de los cambios realizados entre las dos primeras iteraciones de prototipado se muestra en la [Figura 21](#). En la primera iteración de prototipado, la subida de recursos se realizaba una vez se había accedido a una sala. Sin embargo, se pensó que los recursos debían estar asociados al usuario antes de ser utilizados en una sala, o *workspace*, por lo que esa funcionalidad se trasladó a una pestaña específica que contuviese toda la información de los recursos, tal como se apuntaba en la [Figura 20](#).

Otro de los cambios realizados entre la primera y la segunda iteración de prototipado fue la inclusión de etiquetas para los recursos. En la primera iteración se consideró la creación de etiquetas para los diseños. Sin embargo, al eliminar el módulo de diseños del primer prototipo, se trasladó la creación de etiquetas al módulo de recursos, tal como se muestra en la [Figura 22](#).



Figura 21: Prototipo de la interfaz de subida de recursos. Primera iteración (izquierda) y segunda iteración (derecha).



Figura 22: Prototipo de la interfaz de gestión de etiquetas. Segunda iteración.

En cuanto a los cambios realizados entre la segunda iteración de prototipado y la aplicación final, uno de ellos es que en la aplicación final o, lo que es lo mismo, en la tercera iteración de prototipado, se decidió añadir más tipos de recursos, de manera que el usuario pudiera subir imágenes, vídeos, sonidos y textos, como se muestra en la [Figura 23](#). Esto también influyó en la interfaz del módulo de recursos, ya que se hizo necesario mostrar el tipo de recurso en la pestaña de recursos.

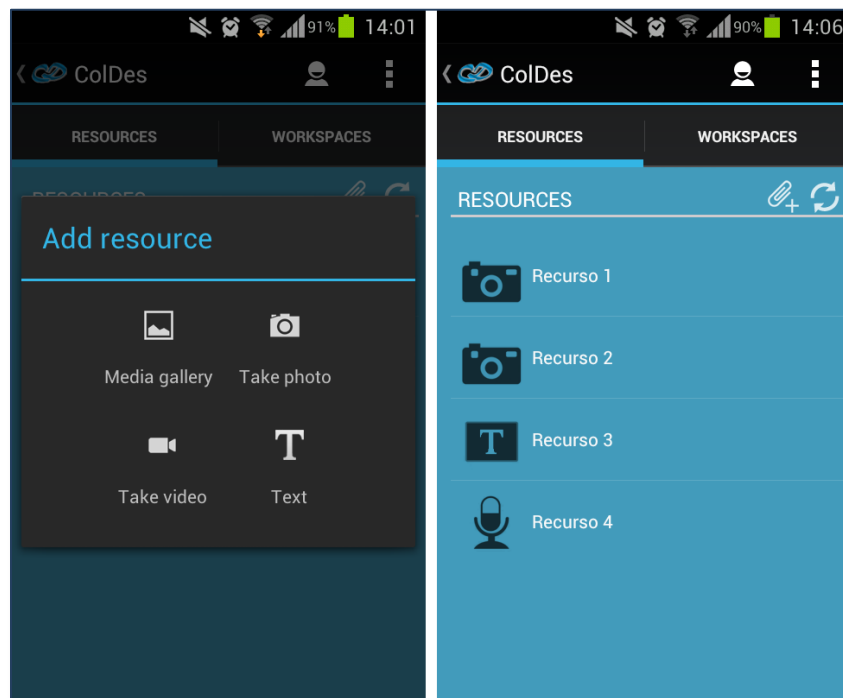


Figura 23: Interfaz de subida de recursos. Aplicación final.

En el módulo de etiquetas, por su parte, se decidió no mostrar la lista de recursos asociado a cada etiqueta en la propia interfaz de etiquetas, sino que esta lista aparecería al pulsar sobre una de las etiquetas. Este cambio se puede observar en la [Figura 24](#).

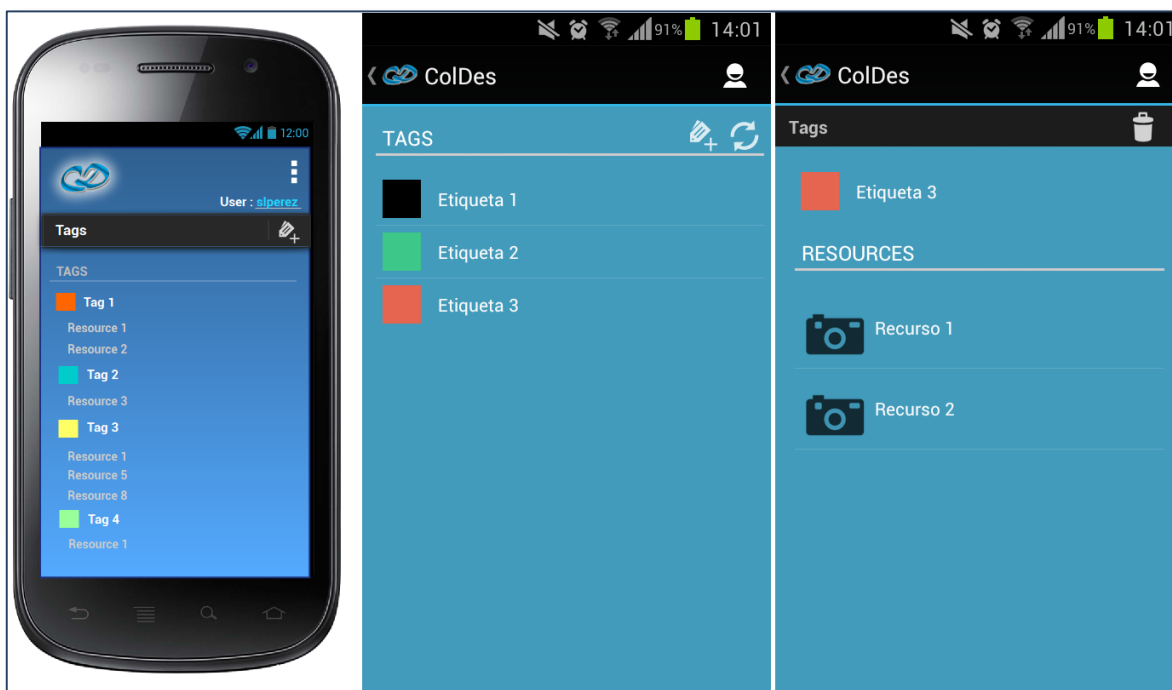


Figura 24: Prototipo de la interfaz del módulo de etiquetas. Segunda iteración (izquierda) y aplicación final (derecha).

Otro de los cambios realizados entre la segunda iteración de prototipado y la aplicación final es la eliminación de la visualización del contenido de un *workspace* tal como se muestra en la [Figura 25](#). Se pensó que, dado que los recursos están asociados a ítems de información dentro de un *workspace*, no tenía sentido asociarlo únicamente al *workspace*. Por otra parte, en un *workspace* se puede asociar el mismo recurso a varios ítems, por lo que no tendría sentido mostrar una lista de recursos duplicados.



Figura 25: Prototipo de la interfaz del módulo de *workspace*. Segunda iteración.

Implementación

En este apartado se presenta la organización del proyecto, así como las herramientas utilizadas para la implementación del sistema. Para ello, se distinguen las dos aplicaciones implementadas: la aplicación web y la aplicación móvil, así como el desarrollo del *Web service*.

Entorno de desarrollo

Tanto para el desarrollo de la aplicación web como para el desarrollo de la aplicación móvil el entorno de desarrollo utilizado es Eclipse [24]. Se trata de un entorno de desarrollo integrado (IDE) multiplataforma y libre que permite crear aplicaciones a partir de un editor de código con resaltado de sintaxis y con compilación en tiempo real. Es utilizado principalmente para crear aplicaciones Java, pero permite la realización de aplicaciones en otros lenguajes únicamente añadiendo el *plug-in* adecuado.

Uno de los *plug-in* desarrollados para Eclipse es el *Adobe Flex Builder Professional Eclipse Plug-in* [25]. Se trata de un módulo para Eclipse que permite al desarrollador realizar proyectos para la plataforma *Flash*. Tras instalar y configurar este *plug-in* en Eclipse, se realiza una adaptación del entorno *Flash Builder* [26] en Eclipse, y es posible crear código *Flash* y *ActionScript* para aplicaciones web (*Flash*) y de escritorio (*AIR*). Este *plug-in* no es libre, por lo que es necesario adquirirlo desde la página web de Adobe. En este caso, se ha utilizado una licencia de estudiante para permitir el desarrollo de la aplicación web.

Otro de los *plug-in* desarrollados para Eclipse es el *Android Development Tools* (ADT) [27], que permite la creación de aplicaciones Android en el IDE Eclipse. Permite configurar de manera rápida nuevos proyectos Android, crear la interfaz de usuario de la aplicación, depurar las aplicaciones utilizando las herramientas del SDK de Android, y exportar las aplicaciones a formato *.apk*, de forma que puedan ser distribuidas en el mercado de aplicaciones de Android. Al igual que el SDK de Android, este *plug-in* es completamente libre. Se ha utilizado en el presente proyecto para la creación de la aplicación móvil. En la [Figura 26](#) se muestra el proyecto de la aplicación móvil en entorno de desarrollo Eclipse, utilizando el *plug-in* ADT.

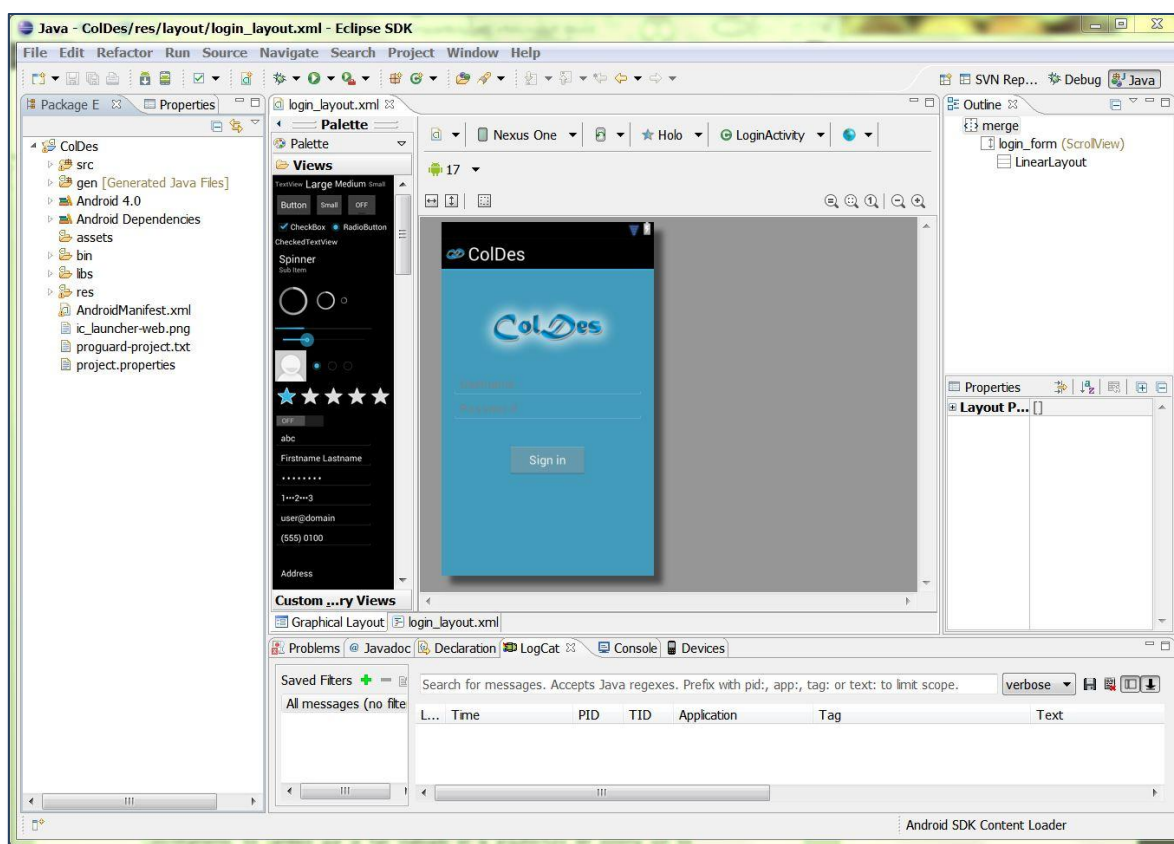


Figura 26: IDE Eclipse con ADT *plug-in*.

En cuanto al *Web service*, se ha utilizado el entorno de desarrollo NetBeans [28] para su desarrollo. Se trata de un IDE libre utilizado principalmente para crear aplicaciones Java. El motivo por el que se ha utilizado NetBeans para la creación del *Web service*, y no Eclipse, es porque ofrece numerosas facilidades para la creación de servicios web. Principalmente, permite utilizar una sintaxis específica para la creación de *Web services* en lenguaje Java. Además, permite configurar fácilmente el servidor Apache Tomcat para el despliegue y pruebas del servicio web. En la [Figura 27](#) se muestra el proyecto del *Web service* en el IDE NetBeans.

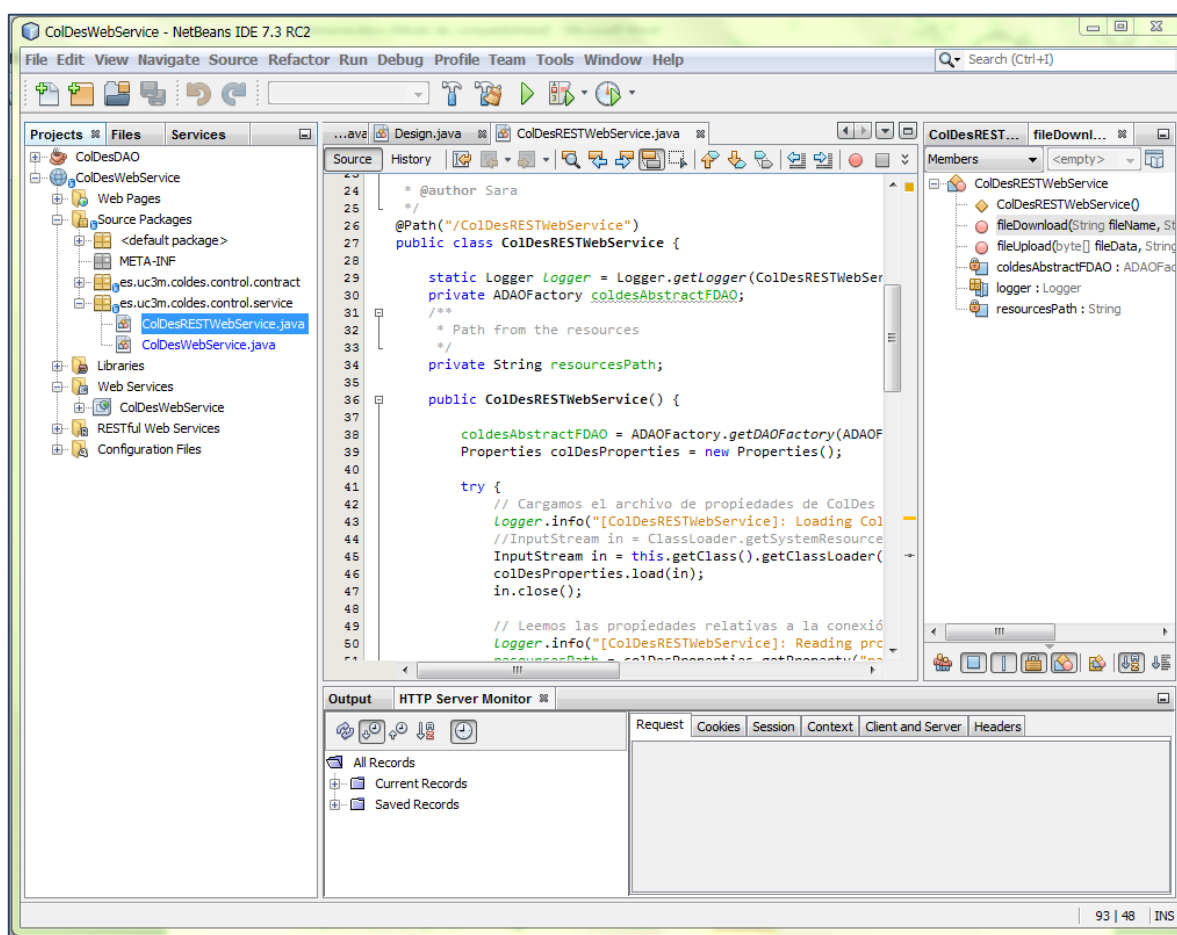


Figura 27: IDE NetBeans.

Organización del proyecto

En las páginas siguientes, se muestra la organización del proyecto tanto para la aplicación móvil y la aplicación web, como para el *Web service*. Para las dos primeras, se ha utilizado el IDE Eclipse, mientras que para el segundo se ha utilizado el IDE NetBeans.

Aplicación web

El proyecto en el que se ha desarrollado la aplicación web parte del proyecto de la aplicación original ColDes, por lo que la estructura del proyecto es prácticamente igual a la que se presentó en la memoria del proyecto fin de carrera de ColDes [1]. La organización del nuevo proyecto se muestra en la *Figura 28*.

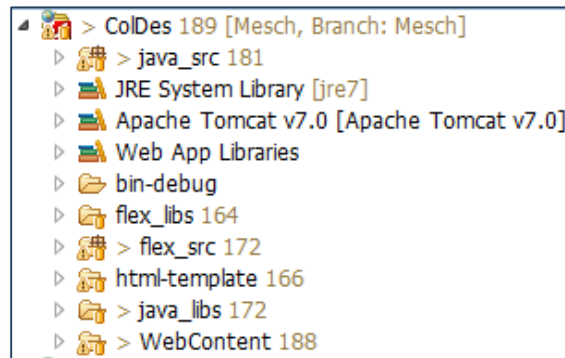


Figura 28: Organización general del proyecto para la aplicación web.

En el directorio *java_src* se encuentran los paquetes con el código de la parte servidora de la aplicación web. Como se puede observar en la [Figura 29](#), se ha simplificado notablemente la estructura de paquetes con respecto al proyecto original. En concreto, se han eliminado por completo los paquetes *es.uc3m.coldes.dao* y *es.uc3m.coldes.dto*, que contenían los DAO y DTO del proyecto, respectivamente. El motivo por el que se han eliminado es porque la comunicación se realiza ahora a través del *middleware*, y es en éste en el que se encuentran las clases correspondientes con los DAO y DTO necesarios para realizar la comunicación con la capa de persistencia. En el *middleware*, por su parte, se encuentra en el proyecto en forma de librería en formato *.jar* (véase apartado [Librería ColDesDAO](#)).

La organización final del directorio *java_src* es la siguiente:

- ***es.uc3m.coldes.control.client*:** Contiene las clases que realizan las llamadas al *middleware* y devuelven el resultado obtenidos a la parte cliente de la aplicación.
- ***es.uc3m.coldes.control.security*:** Contiene la clase *SHA1.java*, que permite encriptar las contraseñas de los usuarios antes de ser añadidas a la base de datos.
- ***es.uc3m.coldes.control.server*:** Contiene las clases que definen la interfaz de las clases del paquete *es.uc3m.control.client*.
- ***es.uc3m.coldes.control.exception*:** Contiene la excepción personalizada que se lanza cuándo expira la sesión de un usuario.
- ***es.uc3m.coldes.control.utils*:** Contiene otras clases de interés para el desarrollo de la parte servidora de la aplicación web, como clases para la lectura de ficheros externos.

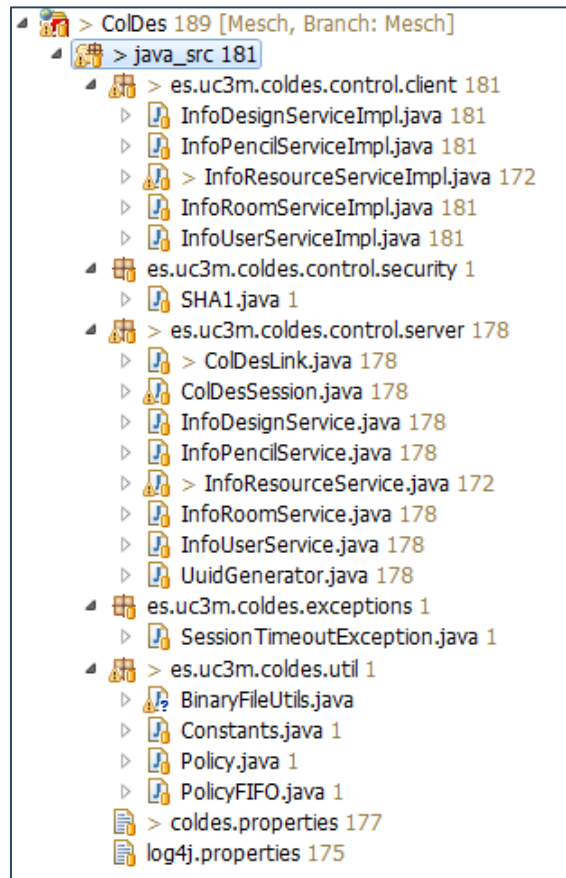


Figura 29: Organización del proyecto para la aplicación web. Parte servidora.

Por otro lado, en el directorio *flex_src* se encuentran los paquetes con el código de la parte cliente de la aplicación web. Se trata de las clases, módulos y componentes se permiten crear la interfaz de usuario de la aplicación web. Como se puede observar en [Figura 30](#), la estructura de paquetes se mantiene con respecto a la aplicación original, tal como se define a continuación:

- **assets:** Almacena los recursos utilizados por la aplicación. En este caso, se trata de los iconos e imágenes utilizados en la interfaz de usuario, organizados en distintos paquetes.
- **es.uc3m.coldes.business:** Contiene las clases que permiten la comunicación con la parte servidora de la aplicación web.
- **es.uc3m.coldes.entities:** Contiene las entidades del modelo de información.
- **es.uc3m.coldes.events:** Contiene la clase con los eventos personalizados para la creación de ítems y grupos de ítems.
- **es.uc3m.coldes.module:** Contiene la definición de los módulos de interfaz de la aplicación.

- **es.uc3m.coldes.renderer:** Contiene los componentes necesarios para crear listas de objetos personalizadas.
- **es.uc3m.coldes.skin:** Contiene los ficheros que permiten la personalización de la interfaz de elementos originales de las librerías de *Flex*.
- **es.uc3m.coldes.utils:** Contiene otras clases de interés para el desarrollo de la parte cliente de la aplicación web, como interfaces personalizadas o clases para la lectura de ficheros externos.
- **es.uc3m.coldes.view:** Contiene los ficheros de definición de interfaz de elementos concretos. Se encuentran organizados en distintos paquetes, según el módulo de interfaz al que correspondan.

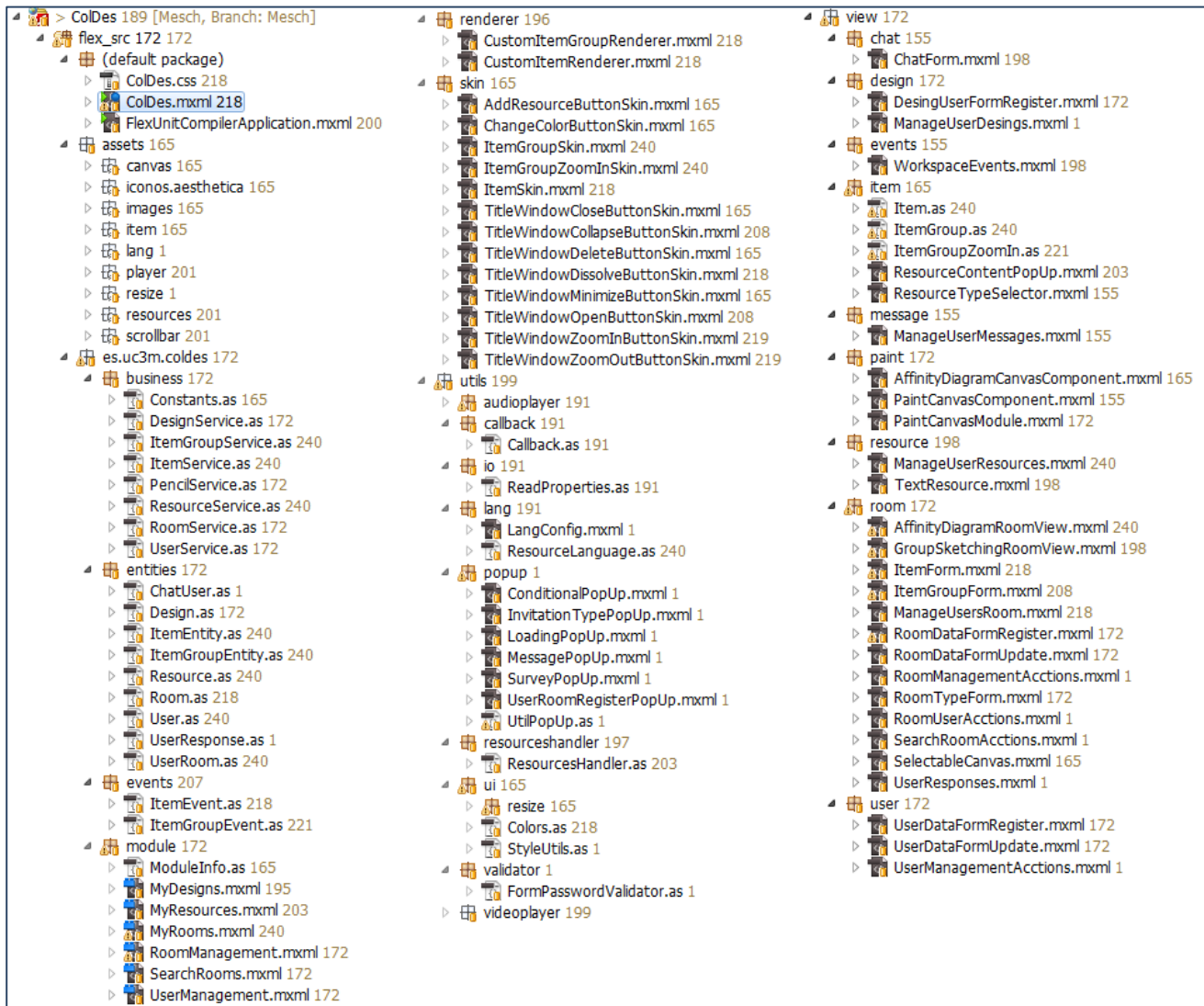


Figura 30: Organización del proyecto para la aplicación web. Parte cliente.

Aplicación móvil

La aplicación móvil se ha desarrollado en el IDE Eclipse, utilizando el *plug-in* ADT. La organización general del proyecto se corresponde a la organización general de los proyectos Android, tal como se muestra en la [Figura 31](#). El contenido de los directorios más importantes es el siguiente:

- **Directorio *src*:** Contiene el código fuente de la aplicación estructurado en varios paquetes que permiten organizar las clases *.java*.
- **Directorio *gen*:** Contiene los archivos generados por el compilador.
- **Directorio *assets*:** Almacena los recursos utilizados por la aplicación, como ficheros de audio o de texto.
- **Directorio *res*:** Contiene los recursos utilizados por la aplicación, como iconos, cadenas de valores, ficheros de interfaz de usuario, etc.

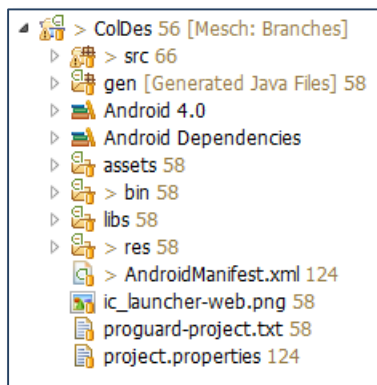


Figura 31: Organización general del proyecto para la aplicación móvil.

Como se acaba de mostrar, el directorio *src* contiene los paquetes y clases *.java* cuyo código define la funcionalidad de la aplicación. La estructura de este directorio se muestra en la [Figura 32](#). El contenido de cada uno de los paquetes es el siguiente:

- ***es.uc3m.coldes*:** Contiene las *activities* de Android que se asocian con la interfaz de usuario y que, por tanto, permiten la interacción con el usuario. Se encargan de cargar la interfaz de usuario correspondiente y definir su comportamiento.
- ***es.uc3m.interaction*:** Las clases de este paquete permiten capturar los eventos del usuario sobre la pantalla del dispositivo móvil.
- ***es.uc3m.coldes.model*:** Contiene las entidades del modelo de información.
- ***es.uc3m.coldes.semantics*:** Las clases de este paquete permiten definir el comportamiento de los eventos del usuario capturados sobre la pantalla del dispositivo móvil.

- **es.uc3m.coldes.adapters:** Contiene las clases necesarias para crear listas de objetos personalizadas, como listas de recursos, listas de *workspaces*, o listas en forma de cuadrícula.
- **es.uc3m.coldes.asynctasks:** Contiene las tareas asíncronas que permiten realizar las operaciones de comunicación de la aplicación con el *middleware*.
- **es.uc3m.coldes.handlers:** Contiene las clases que permiten manejar ciertas acciones del usuario, mostrando un cuadro de diálogo con la información requerida.
- **es.uc3m.coldes.utils:** Contiene otras clases de interés para el desarrollo de la aplicación móvil.

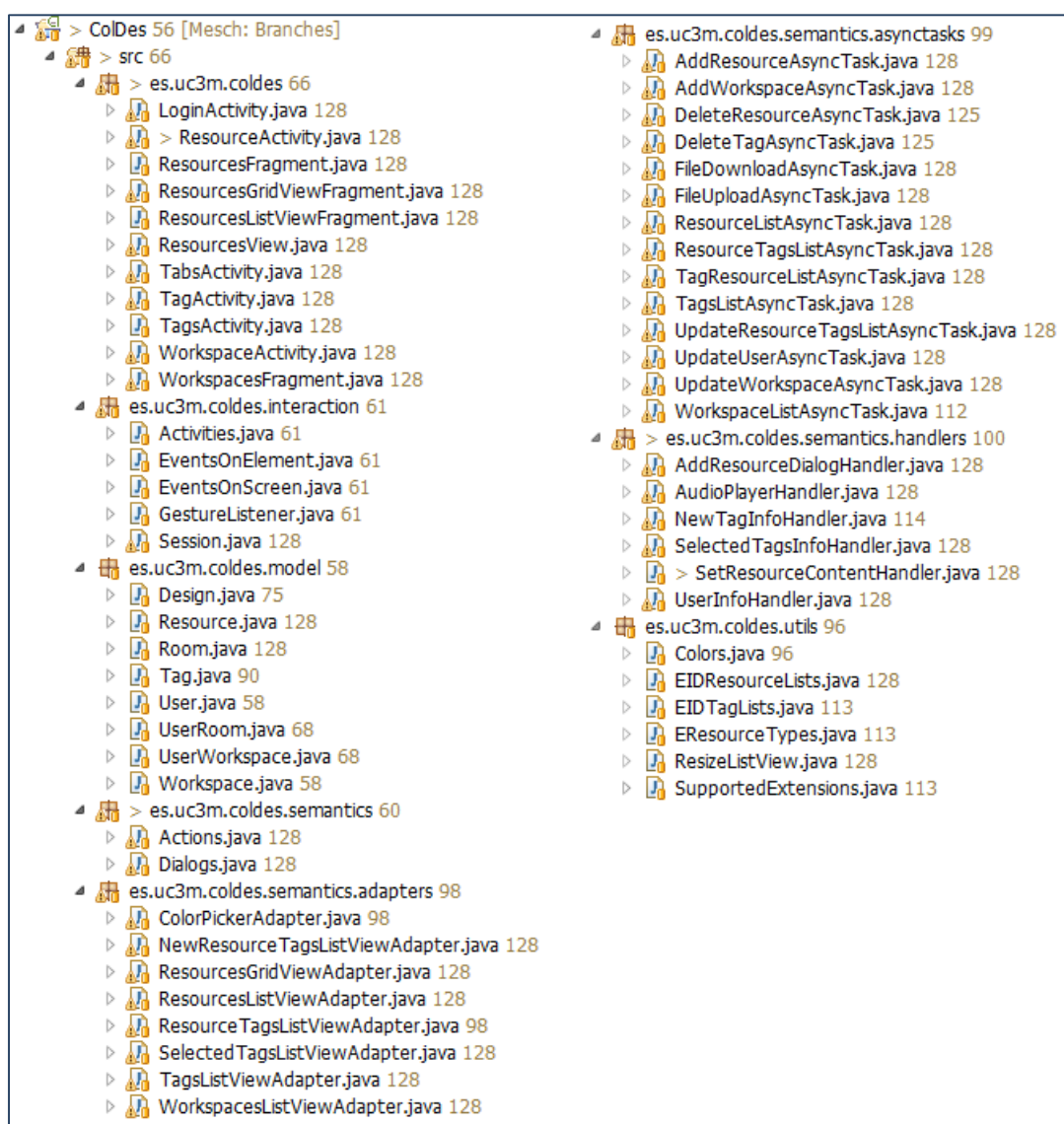


Figura 32: Organización del proyecto para la aplicación móvil. Directorio *src*.

En cuanto al directorio *res*, contiene los recursos que se utilizan en la aplicación. Este directorio contiene, a su vez, distintos directorios que permiten organizar los recursos (véase [Figura 33](#)). Estos directorios son los siguientes:

- ***drawable***: Este directorio contiene los ficheros de imagen o *.xml* que sirven para describir los contenidos que se muestran como imagen en la aplicación. Así, se encuentran los iconos o los ficheros de definición de elementos visuales, como bordes, cabeceras, etc. Existen varios directorios que extienden de éste. Se trata de *drawable-hdpi*, *drawable-ldpi*, *drawable-mdpi* y *drawable-xhdpi*. Estos directorios contienen los recursos que se utilizan según la densidad de píxeles de la pantalla del dispositivo en el que se ejecuta la aplicación: alta, baja, media o extra alta, respectivamente [29].
- ***layout***: Contiene los ficheros *.xml* que definen la interfaz de usuario de la aplicación.
- ***menu***: Contiene los ficheros *.xml* que definen los menús de la aplicación.
- ***values***: Contiene los ficheros que permiten definir los estilos, cadenas de texto, colores, etc. de la aplicación. Existen varios directorios que pueden extender de éste. Por ejemplo, en el fichero *strings.xml* del directorio *values-es* se definen las cadenas de texto que se deben utilizar si el idioma del dispositivo es español, mientras que en el fichero *strings.xml* del directorio *values* se definen las cadenas de texto que se muestran por defecto, en este caso, en inglés.

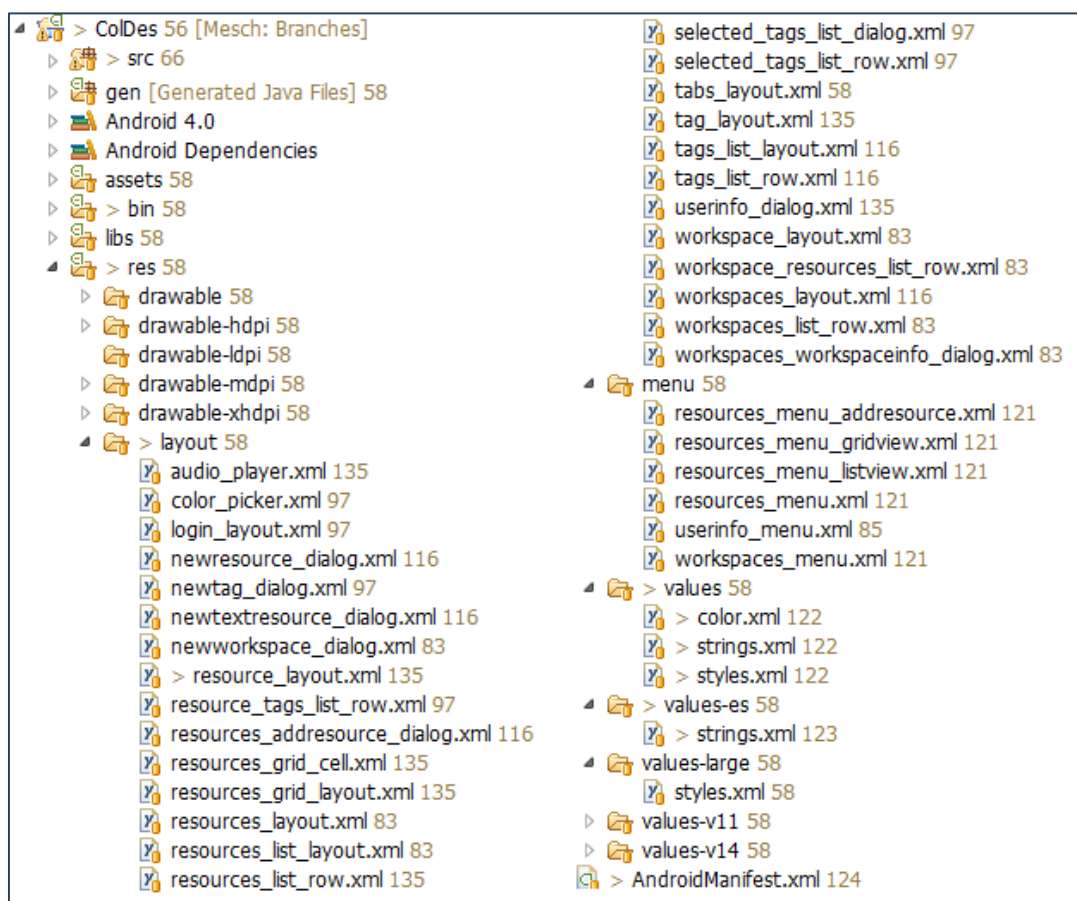


Figura 33: Organización del proyecto para la aplicación móvil. Directorio *res*.

Web service

La organización del proyecto en el que se desarrolla el *Web service* que forma parte del *middleware* se muestra en la [Figura 34](#). Este proyecto se ha desarrollado en el IDE NetBeans y cuenta con los siguientes paquetes:

- ***es.uc3m.coldes.control.contract***: Contiene la interfaz que debe implementar el *Web service* SOAP.
- ***es.uc3m.coldes.control.service***: Contiene las clases que implementan los *Web services* para los protocolos SOAP y REST. La clase *ColDesWebService.java* se corresponde con el protocolo SOAP y contiene todos aquellos métodos que permiten al cliente guardar u obtener datos de la base de datos. La clase *ColDesRESTWebService.java* se corresponde con el protocolo REST y contiene los métodos que permiten guardar y obtener recursos del sistema de ficheros.

Por otra parte, en el directorio *Libraries* se encuentra el fichero *ColDesDAO.jar*, la librería que contiene los ficheros correspondientes a los DAO y DTO del proyecto.

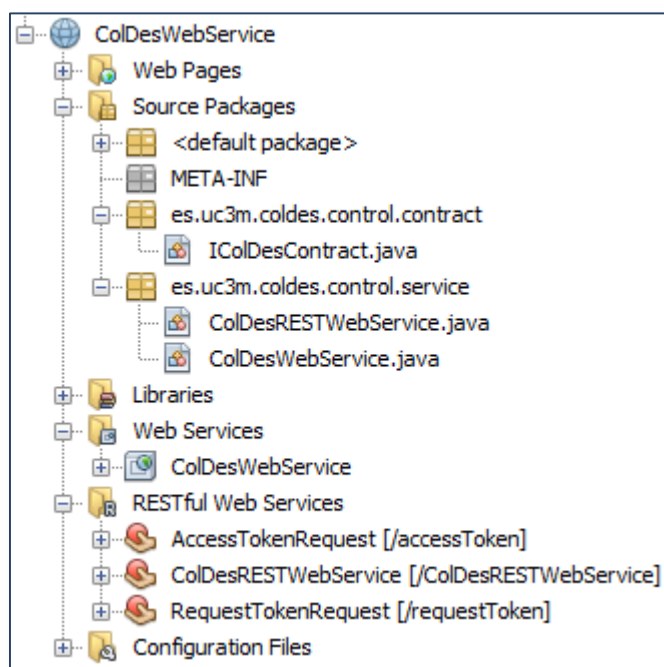


Figura 34: Organización del proyecto para el Web service.

Librería ColDesDAO

Como se ha especificado líneas arriba, se ha eliminado del proyecto de la aplicación web los ficheros correspondientes a los DAO y DTO del proyecto para incorporarlos en una librería que fuese accesible a través del Web service a partir de la aplicación web y la aplicación móvil. La organización de esta librería se muestra en la [Figura 35](#). En la [Figura 17](#) se muestra un extracto del diagrama de clases correspondiente a la librería.

- **es.uc3m.coldes.control.dao:** Contiene las clases correspondientes a los participantes *Abstract Factory* y *Concrete Factory* del patrón *AbstractFactory* (véase el apartado [Diseño detallado](#)).
- **es.uc3m.coldes.control.dao.coldes:** Contiene las clases correspondientes al participante *Abstract Product* del patrón *AbstractFactory*. Se trata de las interfaces que definen las operaciones que es posible realizar para cada entidad en la base de datos.
- **es.uc3m.coldes.control.dao.coldes.mysql:** Contiene las clases correspondientes al participante *Concret Product* del patrón *AbstractFactory*, para el caso de utilizar una base de datos MySQL en la capa de persistencia. En estas clases se encuentran las implementaciones de las clases *Abstract Product*.
- **es.uc3m.coldes.model:** Contiene las clases correspondientes a los DTO del proyecto.

- ***es.uc3m.coldes.security***: Contiene la clase necesaria para codificar las contraseñas de los usuarios antes de insertarlas en la base de datos.
- ***es.uc3m.coldes.util***: Contiene otras clases de interés para el desarrollo del proyecto.

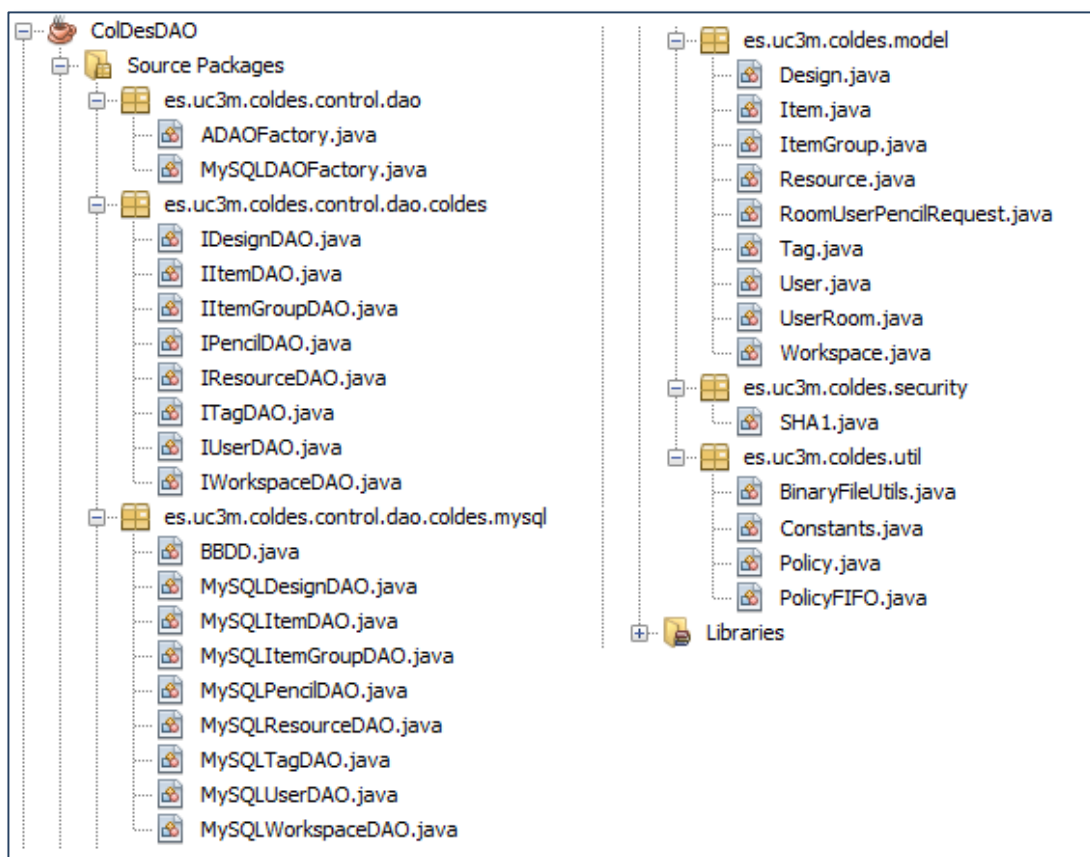


Figura 35: Organización de la librería ColDesDAO.

5 Evaluación

La fase de evaluación de un proyecto software tiene como objetivo demostrar la validez de la solución elaborada. Para que la solución aportada se considere válida debe resolver los problemas expuestos en la sección *Problem definition*, así como satisfacer los objetivos definidos en la sección *Aims*.

En este capítulo se evalúa si el sistema desarrollado es válido a través de la realización de una serie de pruebas de validación. Para ello se presenta, en primer lugar, el proceso de evaluación llevado a cabo para validar el sistema. Éste incluye la definición del plan de pruebas a realizar y los casos de prueba que permiten validar el sistema. A partir de la realización de estas pruebas se presenta, en segundo lugar, el análisis de los resultados obtenidos.

5.1 Proceso de evaluación

En la presente sección se define la forma en que el sistema se ha evaluado, así como los casos de prueba realizados para la evaluación.

Plan de pruebas

En el presente apartado se expone el plan de pruebas utilizado para evaluar el sistema. Se ha decidido realizar distintos casos de prueba que permitan verificar que el sistema implementado cumple con los requisitos establecidos en la fase de análisis. En concreto, los casos de prueba estarán relacionados con los casos de uso presentados en el apartado *Casos de uso*. Puesto que todos los requisitos están recogidos en estos casos de uso (véase *Tabla 12* y *Tabla 13*), los casos de prueba permiten establecer si los requisitos se cumplen en el sistema desarrollado.

Los casos de prueba están definidos según la siguiente plantilla [30]:

Identificador	PR-<Tipo de aplicación>-<Numero>
Módulo	
Casos de uso	
Precondiciones	
Procedimiento	
Verificación	

Tabla 22: Plantilla de especificación de casos de prueba.

El significado de cada uno de los campos de la plantilla de casos de prueba es el siguiente:

- **Identificador:** Identificador del caso de prueba, compuesto por los siguientes campos:
 - **Tipo de aplicación:** Diferencia entre los casos de prueba dedicados a la evaluación de la aplicación web y los requisitos dedicados a la evaluación de la aplicación móvil, tomando los valores “W” y “M” respectivamente.
 - **Número:** Número que identifica de manera unívoca el caso de prueba.
- **Módulo:** Módulo de funcionalidad al que pertenece el caso de prueba.
- **Casos de uso:** Identificador del caso de uso (o casos de uso) que es posible verificar con el caso de prueba.
- **Precondiciones:** Condiciones que se deben cumplir para que el procedimiento del caso de prueba se pueda llevar a cabo.
- **Procedimiento:** Secuencia de acciones que es necesario realizar para verificar el caso de prueba (o casos de prueba) dado por el campo anterior.
- **Verificación:** Resultado esperado tras la realización de la secuencia de acciones para el caso en que el caso de uso (o casos de uso) se verifique.

Casos de prueba

A partir de la plantilla presentada en la [Tabla 22](#) se crean todos los casos de prueba necesarios para verificar los casos de uso del apartado [Casos de uso](#). Estos casos de uso se corresponden con los escenarios que es necesario evaluar en el sistema, ya que definen la funcionalidad total del mismo. Si es posible verificar los casos de uso, se puede validar el sistema.

En la mayoría de los casos, un caso de prueba sirve para verificar un único caso de uso, pero existen casos en los que sirve para verificar más de un caso de uso.

A continuación se presentan los casos de prueba necesarios para evaluar la aplicación web:

Identificador	PR-W-1
Módulo	Crear <i>workspace</i> para la creación de diagramas de afinidad
Casos de uso	CU-W-1
Precondiciones	– El usuario debe haberse autenticado en la aplicación web ColDes.
Procedimiento	<ol style="list-style-type: none"> 1. Acceder al espacio “Mis <i>workspaces</i>”. 2. Pulsar sobre el botón “crear workspace”. 3. Elegir la opción de crear un <i>workspace</i> para la creación de diagramas de afinidad. 4. Introducir el nombre del <i>workspace</i>. 5. Introducir como información de colaboración “Uno a uno”. 6. Pulsar sobre el botón “Aceptar”. 7. Acceder al <i>workspace</i> recién creado.

	8. Solicitar el testigo para realizar operaciones en el <i>workspace</i> . 9. Realizar operaciones en el <i>workspace</i> .
Verificación	<ul style="list-style-type: none"> – Aparece un cuadro de verificación que indica que la operación ha sido realizada con éxito. – Se refresca la lista de <i>workspaces</i> del usuario y aparece el <i>workspace</i> recién creado. – Sólo es posible realizar operaciones en el <i>workspace</i> cuando se ha obtenido el testigo. – Se crea un nuevo registro en la tabla “room” de la base de datos con los datos del <i>workspace</i> recién creado.

Identificador	PR-W-2
Módulo	Crear <i>workspace</i> para la creación de diagramas de afinidad
Casos de uso	CU-W-1
Precondiciones	<ul style="list-style-type: none"> – El usuario debe haberse autenticado en la aplicación web ColDes.
Procedimiento	<ol style="list-style-type: none"> 1. Acceder al espacio “Mis <i>workspaces</i>”. 2. Pulsar sobre el botón “crear workspace”. 3. Elegir la opción de crear un <i>workspace</i> para la creación de diagramas de afinidad. 4. Introducir el nombre del <i>workspace</i>. 5. Introducir como información de colaboración “Todos a la vez”. 6. Pulsar sobre el botón “Aceptar”. 7. Acceder al <i>workspace</i> recién creado. 8. Realizar operaciones en el <i>workspace</i>.
Verificación	<ul style="list-style-type: none"> – Aparece un cuadro de verificación que indica que la operación ha sido realizada con éxito. – Se refresca la lista de <i>workspaces</i> del usuario y aparece el <i>workspace</i> recién creado. – No es necesario solicitar un testigo para realizar operaciones en el <i>workspace</i>. – Se crea un nuevo registro en la tabla “room” de la base de datos con los datos del <i>workspace</i> recién creado.

Identificador	PR-W-3
Módulo	Crear ítems de información
Casos de uso	CU-W-2
Precondiciones	<ul style="list-style-type: none"> – El usuario debe haberse autenticado en la aplicación web ColDes.

	– PR-W-1 o PR-W-2
Procedimiento	<ol style="list-style-type: none"> 1. Acceder al espacio “Mis <i>workspaces</i>”. 2. Abrir un <i>workspace</i> para la creación de diagramas de afinidad. 3. Pulsar sobre el botón de creación de ítems en el <i>ribbon</i> del <i>workspace</i>. 4. Introducir como texto asociado al ítem de información “Nuevo ítem de información”. 5. Pulsar sobre el botón “Aceptar”.
Verificación	<ul style="list-style-type: none"> – En el espacio de trabajo del <i>workspace</i> aparece un recuadro con aspecto similar a una nota adhesiva. Esta nota incluye el texto “Nuevo ítem de información”. – Se crea un nuevo registro en la tabla “roomitem” de la base de datos con los datos del ítem recién creado.

Identificador	PR-W-4
Módulo	Crear ítems de información
Casos de uso	CU-W-2
Precondiciones	<ul style="list-style-type: none"> – El usuario debe haberse autenticado en la aplicación web ColDes. – PR-W-1 o PR-W-2
Procedimiento	<ol style="list-style-type: none"> 1. Acceder al espacio “Mis <i>workspaces</i>”. 2. Abrir un <i>workspace</i> para la creación de diagramas de afinidad. 3. Pulsar sobre el botón de creación de ítems en el <i>ribbon</i> del <i>workspace</i>. 4. Dejar vacío el cuadro en el que se introduce el texto del ítem de información. 5. Pulsar sobre el botón “Aceptar”.
Verificación	<ul style="list-style-type: none"> – Se muestra un mensaje de error indicando que el cuadro de texto no puede estar vacío. – No se realizan cambios en la base de datos.

Identificador	PR-W-5
Módulo	Personalizar ítems de información
Casos de uso	CU-W-4
Precondiciones	<ul style="list-style-type: none"> – El usuario debe haberse autenticado en la aplicación web ColDes. – PR-W-1 o PR-W-2 – PR-W-4
Procedimiento	<ol style="list-style-type: none"> 1. Acceder al espacio “Mis <i>workspaces</i>”. 2. Abrir un <i>workspace</i> para la creación de diagramas de afinidad. 3. Pulsar sobre un ítem de información.

	4. Posicionar el ítem arrastrando la parte superior del ítem a una posición del espacio de trabajo.
Verificación	<ul style="list-style-type: none"> – El ítem de información se desplaza a la posición indicada. – Se actualiza el registro de la tabla “roomitem” de la base de datos con los datos de posición del ítem.

Identificador	PR-W-6
Módulo	Personalizar ítems de información
Casos de uso	CU-W-4
Precondiciones	<ul style="list-style-type: none"> – El usuario debe haberse autenticado en la aplicación web ColDes. – PR-W-1 o PR-W-2 – PR-W-4
Procedimiento	<ol style="list-style-type: none"> 1. Acceder al espacio “Mis <i>workspaces</i>”. 2. Abrir un <i>workspace</i> para la creación de diagramas de afinidad. 3. Pulsar sobre un ítem de información. 4. Posicionar el ítem arrastrando la parte superior del ítem a una posición fuera del espacio de trabajo.
Verificación	<ul style="list-style-type: none"> – El ítem de información se desplaza hasta la posición límite del espacio de trabajo, y no se permite posicionar el ítem fuera de este espacio. – Se actualiza el registro de la tabla “roomitem” de la base de datos con los datos de posición del ítem.

Identificador	PR-W-7
Módulo	Personalizar ítems de información
Casos de uso	CU-W-3
Precondiciones	<ul style="list-style-type: none"> – El usuario debe haberse autenticado en la aplicación web ColDes. – PR-W-1 o PR-W-2 – PR-W-4
Procedimiento	<ol style="list-style-type: none"> 1. Acceder al espacio “Mis <i>workspaces</i>”. 2. Abrir un <i>workspace</i> para la creación de diagramas de afinidad. 3. Pulsar sobre un ítem de información. 4. Redimensionar el ítem arrastrando la esquina inferior derecha del ítem a una posición del espacio de trabajo.
Verificación	<ul style="list-style-type: none"> – El ítem de información se redimensiona a la posición indicada. – Se actualiza el registro de la tabla “roomitem” de la base de datos con los nuevos datos de dimensión del ítem.

Identificador	PR-W-8
Módulo	Personalizar ítems de información
Casos de uso	CU-W-3
Precondiciones	<ul style="list-style-type: none"> – El usuario debe haberse autenticado en la aplicación web ColDes. – PR-W-1 o PR-W-2 – PR-W-4
Procedimiento	<ol style="list-style-type: none"> 1. Acceder al espacio “Mis <i>workspaces</i>”. 2. Abrir un <i>workspace</i> para la creación de diagramas de afinidad. 3. Pulsar sobre un ítem de información. 4. Redimensionar el ítem arrastrando la esquina inferior derecha del ítem a una posición fuera del espacio de trabajo.
Verificación	<ul style="list-style-type: none"> – El ítem de información se redimensiona hasta la posición límite del espacio de trabajo, y no se permite redimensionar el ítem fuera de este espacio. – Se actualiza el registro de la tabla “roomitem” de la base de datos con los nuevos datos de dimensión del ítem.

Identificador	PR-W-9
Módulo	Personalizar ítems de información
Casos de uso	CU-W-3
Precondiciones	<ul style="list-style-type: none"> – El usuario debe haberse autenticado en la aplicación web ColDes. – PR-W-1 o PR-W-2 – PR-W-4
Procedimiento	<ol style="list-style-type: none"> 1. Acceder al espacio “Mis <i>workspaces</i>”. 2. Abrir un <i>workspace</i> para la creación de diagramas de afinidad. 3. Pulsar sobre un ítem de información. 4. Pulsar sobre el selector de color del ítem y escoger el nuevo color del ítem.
Verificación	<ul style="list-style-type: none"> – El ítem de información adopta el nuevo color escogido. – Se actualiza el registro de la tabla “roomitem” de la base de datos con el nuevo color del ítem.

Identificador	PR-W-10
Módulo	Eliminar ítems de información
Casos de uso	CU-W-4
Precondiciones	<ul style="list-style-type: none"> – El usuario debe haberse autenticado en la aplicación web ColDes.

	<ul style="list-style-type: none"> – PR-W-1 o PR-W-2 – PR-W-4
Procedimiento	<ol style="list-style-type: none"> 1. Acceder al espacio “Mis <i>workspaces</i>”. 2. Abrir un <i>workspace</i> para la creación de diagramas de afinidad. 3. Pulsar sobre un ítem de información. 4. Pulsar sobre el botón del ítem de información que permite eliminarlo. 5. Aceptar el mensaje de confirmación.
Verificación	<ul style="list-style-type: none"> – El ítem de información desaparece del espacio de trabajo del <i>workspace</i>. – Se elimina el registro de la tabla “roomitem” de la base de datos correspondiente al ítem seleccionado, así como los registros de las tablas intermedias “itemgroup” y “ítemresource” correspondientes.

Identificador	PR-W-11
Módulo	Eliminar ítems de información
Casos de uso	CU-W-4
Precondiciones	<ul style="list-style-type: none"> – El usuario debe haberse autenticado en la aplicación web ColDes. – PR-W-1 o PR-W-2 – PR-W-4
Procedimiento	<ol style="list-style-type: none"> 1. Acceder al espacio “Mis <i>workspaces</i>”. 2. Abrir un <i>workspace</i> para la creación de diagramas de afinidad. 3. Pulsar sobre un ítem de información. 4. Pulsar sobre el botón del ítem de información que permite eliminarlo. 5. Cancelar el mensaje de confirmación.
Verificación	<ul style="list-style-type: none"> – El ítem de información permanece el espacio de trabajo del <i>workspace</i>. – No se realizan cambios en la base de datos.

Identificador	PR-W-12
Módulo	Asociar recursos a un ítem de información
Casos de uso	CU-W-5
Precondiciones	<ul style="list-style-type: none"> – El usuario debe haberse autenticado en la aplicación web ColDes. – PR-W-1 o PR-W-2 – PR-W-4
Procedimiento	<ol style="list-style-type: none"> 1. Acceder al espacio “Mis <i>workspaces</i>”. 2. Abrir un <i>workspace</i> para la creación de diagramas de afinidad. 3. Pulsar sobre un ítem de información.

	<ol style="list-style-type: none"> 4. Pulsar sobre el botón del ítem de información que permite asociar un recurso al mismo. 5. Escoger el recurso a asociar. 6. Pulsar sobre el botón “Aceptar”.
Verificación	<ul style="list-style-type: none"> – En la lista de recursos de la parte inferior del ítem de información se añade el recurso seleccionado. – Se crea un nuevo registro en la tabla “itemresource” que relacione el ítem de información con el recurso añadido.

Identificador	PR-W-13
Módulo	Asociar recursos a un ítem de información
Casos de uso	CU-W-5
Precondiciones	<ul style="list-style-type: none"> – El usuario debe haberse autenticado en la aplicación web ColDes. – PR-W-1 o PR-W-2 – PR-W-4
Procedimiento	<ol style="list-style-type: none"> 1. Acceder al espacio “Mis <i>workspaces</i>”. 2. Abrir un <i>workspace</i> para la creación de diagramas de afinidad. 3. Pulsar sobre un ítem de información. 4. Pulsar sobre el botón del ítem de información que permite asociar un recurso al mismo. 5. La lista de recursos está vacía.
Verificación	<ul style="list-style-type: none"> – Aparece un mensaje indicando que no existen recursos asociados a la cuenta de usuario. – No se realizan cambios en la base de datos.

Identificador	PR-W-14
Módulo	Eliminar recursos asociados a un ítem de información
Casos de uso	CU-W-6
Precondiciones	<ul style="list-style-type: none"> – El usuario debe haberse autenticado en la aplicación web ColDes. – PR-W-1 o PR-W-2 – PR-W-4
Procedimiento	<ol style="list-style-type: none"> 1. Acceder al espacio “Mis <i>workspaces</i>”. 2. Abrir un <i>workspace</i> para la creación de diagramas de afinidad. 3. Pulsar sobre un ítem de información. 4. Pulsar sobre el recurso del ítem de información a eliminar. 5. Arrastrar el recurso hasta la papelera que aparece en el ítem de información.

	6. Aceptar el mensaje de confirmación.
Verificación	<ul style="list-style-type: none"> – En la lista de recursos de la parte inferior del ítem de información se elimina el recurso seleccionado. – Se elimina el registro en la tabla “ítemresource” de la base de datos que relaciona el ítem de información con el recurso añadido.

Identificador	PR-W-15
Módulo	Eliminar recursos asociados a un ítem de información
Casos de uso	CU-W-6
Precondiciones	<ul style="list-style-type: none"> – El usuario debe haberse autenticado en la aplicación web ColDes. – PR-W-1 o PR-W-2 – PR-W-4
Procedimiento	<ol style="list-style-type: none"> 1. Acceder al espacio “Mis <i>workspaces</i>”. 2. Abrir un <i>workspace</i> para la creación de diagramas de afinidad. 3. Pulsar sobre un ítem de información. 4. Pulsar sobre el recurso del ítem de información a eliminar. 5. Arrastrar el recurso hasta la papelera que aparece en el ítem de información. 6. Cancelar el mensaje de confirmación.
Verificación	<ul style="list-style-type: none"> – Permanece el recurso en la lista de recursos de la parte inferior del ítem de información. – No se realizan cambios en la base de datos.

Identificador	PR-W-16
Módulo	Agrupar ítems de información
Casos de uso	CU-W-7
Precondiciones	<ul style="list-style-type: none"> – El usuario debe haberse autenticado en la aplicación web ColDes. – PR-W-1 o PR-W-2 – PR-W-4
Procedimiento	<ol style="list-style-type: none"> 1. Acceder al espacio “Mis <i>workspaces</i>”. 2. Abrir un <i>workspace</i> para la creación de diagramas de afinidad. 3. Seleccionar ítems: Pulsar sobre una posición del espacio de trabajo y arrastrar el cursor hasta la posición que englobe a los ítems que se van a agrupar. 4. Pulsar sobre el botón de agrupación de ítems en el <i>ribbon</i> del <i>workspace</i>. 5. Introducir el título del grupo.

	6. Pulsar sobre el botón “Aceptar”.
Verificación	<ul style="list-style-type: none"> – En el espacio de trabajo del <i>workspace</i> aparece un recuadro en el que se incluyen los ítems de información seleccionados. – El título del grupo es el introducido por el usuario. – Se crea un nuevo registro en la tabla “itemgroup” de la base de datos para cada ítem seleccionado, relacionando cada uno de ellos con el grupo recién creado.

Identificador	PR-W-17
Módulo	Agrupar ítems de información
Casos de uso	CU-W-7
Precondiciones	<ul style="list-style-type: none"> – El usuario debe haberse autenticado en la aplicación web ColDes. – PR-W-1 o PR-W-2 – PR-W-4
Procedimiento	<ol style="list-style-type: none"> 1. Acceder al espacio “Mis <i>workspaces</i>”. 2. Abrir un <i>workspace</i> para la creación de diagramas de afinidad. 3. Seleccionar ítems: Pulsar sobre una posición del espacio de trabajo y arrastrar el cursor hasta la posición que englobe a los ítems que se van a agrupar. 4. Pulsar sobre el botón de agrupación de ítems en el <i>ribbon</i> del <i>workspace</i>. 5. Dejar vacío el cuadro en el que se introduce el título del grupo. 6. Pulsar sobre el botón “Aceptar”.
Verificación	<ul style="list-style-type: none"> – Se muestra un mensaje de error indicando que el cuadro de texto para el título del grupo no puede estar vacío. – No se realizan cambios en la base de datos.

Identificador	PR-W-18
Módulo	Agrupar ítems de información
Casos de uso	CU-W-7
Precondiciones	<ul style="list-style-type: none"> – El usuario debe haberse autenticado en la aplicación web ColDes. – PR-W-1 o PR-W-2 – PR-W-4
Procedimiento	<ol style="list-style-type: none"> 1. Acceder al espacio “Mis <i>workspaces</i>”. 2. Abrir un <i>workspace</i> para la creación de diagramas de afinidad. 3. No se selecciona ningún ítem de información. 4. Pulsar sobre el botón de agrupación de ítems en el <i>ribbon</i> del

	<i>workspace</i> .
Verificación	<ul style="list-style-type: none"> – Se muestra un mensaje de error indicando que se debe seleccionar, al menos, un ítem para poder crear un grupo. – No se realizan cambios en la base de datos.

Identificador	PR-W-19
Módulo	Disolver grupo de ítems de información
Casos de uso	CU-W-8
Precondiciones	<ul style="list-style-type: none"> – El usuario debe haberse autenticado en la aplicación web ColDes. – PR-W-1 o PR-W-2 – PR-W-4 – PR-W-16
Procedimiento	<ol style="list-style-type: none"> 1. Acceder al espacio “<i>Mis workspaces</i>”. 2. Abrir un <i>workspace</i> para la creación de diagramas de afinidad. 3. Pulsar sobre un grupo de ítems de información. 4. Pulsar sobre el botón del grupo situado en la esquina superior derecha de éste que permite eliminarlo. 5. Aceptar el mensaje de confirmación.
Verificación	<ul style="list-style-type: none"> – El grupo de ítems desaparece del espacio de trabajo del <i>workspace</i>. – Se eliminan los registros de la tabla “itemgroup” de la base de datos que relacionan cada ítem con el grupo al que pertenecían.

Identificador	PR-W-20
Módulo	Guardar diagrama de afinidad
Casos de uso	CU-W-9
Precondiciones	<ul style="list-style-type: none"> – El usuario debe haberse autenticado en la aplicación web ColDes. – PR-W-1 o PR-W-2 – PR-W-4 (Opcional) – PR-W-16 (Opcional)
Procedimiento	<ol style="list-style-type: none"> 1. Acceder al espacio “<i>Mis workspaces</i>”. 2. Abrir un <i>workspace</i> para la creación de diagramas de afinidad. 3. Realizar cualquier acción sobre el diagrama de afinidad. 4. Pulsar sobre el botón de guardado del diagrama de afinidad en el <i>ribbon</i> del <i>workspace</i>.
Verificación	<ul style="list-style-type: none"> – Aparece un cuadro de verificación que indica que la operación ha sido realizada con éxito. – Se actualizan los registros de las tablas “roomitem”, “itemgroup” y

	“itemresource” de la base de datos con los datos del diagrama de afinidad.
--	--

Identificador	PR-W-21
Módulo	Cargar diagrama de afinidad
Casos de uso	CU-W-9, CU-W-10
Precondiciones	<ul style="list-style-type: none"> – El usuario debe haberse autenticado en la aplicación web ColDes. – PR-W-1 o PR-W-2 – PR-W-4 (Opcional) – PR-W-16 (Opcional)
Procedimiento	<ol style="list-style-type: none"> 1. Acceder al espacio “Mis <i>workspaces</i>”. 2. Abrir un <i>workspace</i> para la creación de diagramas de afinidad. 3. Realizar cualquier acción sobre el diagrama de afinidad. 4. Pulsar sobre el botón de guardado del diagrama de afinidad en el <i>ribbon</i> del <i>workspace</i>. 5. Salir del <i>workspace</i>. 6. Volver a abrir el <i>workspace</i>. 7. Pulsar sobre el botón de cargado del diagrama de afinidad en el <i>ribbon</i> del <i>workspace</i>.
Verificación	<ul style="list-style-type: none"> – El diagrama de afinidad del <i>workspace</i> se actualiza con la información del diagrama de afinidad guardado. – No se realizan cambios en la base de datos.

Identificador	PR-W-22
Módulo	Visualizar recursos
Casos de uso	CU-W-11
Precondiciones	<ul style="list-style-type: none"> – El usuario debe haberse autenticado en la aplicación web ColDes.
Procedimiento	<ol style="list-style-type: none"> 1. Acceder al espacio “Usuario”. 2. Acceder al espacio “Mis recursos”. 3. Pulsar sobre uno de los recursos de la lista de recursos.
Verificación	<ul style="list-style-type: none"> – El contenido del recurso se visualiza en la parte derecha de la zona de recursos.

Identificador	PR-W-23
Módulo	Visualizar recursos
Casos de uso	CU-W-11

Precondiciones	– El usuario debe haberse autenticado en la aplicación web ColDes.
Procedimiento	<ol style="list-style-type: none"> 4. Acceder al espacio “Usuario”. 5. Acceder al espacio “Mis recursos”. 6. La lista de recursos está vacía.
Verificación	<ul style="list-style-type: none"> – Aparece un mensaje indicando que no existen recursos asociados a la cuenta de usuario. – No se realizan cambios en la base de datos.

A continuación se presentan los casos de prueba necesarios para evaluar la aplicación móvil.

Identificador	PR-M-1
Módulo	Autenticación en el sistema
Casos de uso	CU-M-1
Precondiciones	– Haber creado una cuenta en ColDes con nombre de usuario “name” y contraseña “name*pass”.
Procedimiento	<ol style="list-style-type: none"> 1. Acceder a la aplicación móvil ColDes. 2. Introducir “name” como nombre de usuario y “name*pass” como contraseña. 3. Pulsar el botón “Entrar”.
Verificación	– Se accede a la pantalla principal de la aplicación.

Identificador	PR-M-2
Módulo	Autenticación en el sistema
Casos de uso	CU-M-1
Precondiciones	– Haber creado una cuenta en ColDes con nombre de usuario “name” y contraseña “name*pass”.
Procedimiento	<ol style="list-style-type: none"> 1. Acceder a la aplicación móvil ColDes. 2. Introducir “na” como nombre de usuario y “name*pass” como contraseña. 3. Pulsar el botón “Entrar”.
Verificación	– Aparece un mensaje indicando que el usuario “na” no existe en la aplicación.

Identificador	PR-M-3
Módulo	Autenticación en el sistema
Casos de uso	CU-M-1

Precondiciones	– Haber creado una cuenta en ColDes con nombre de usuario “name” y contraseña “name*pass”.
Procedimiento	<ol style="list-style-type: none"> 1. Acceder a la aplicación móvil ColDes. 2. Introducir “name” como nombre de usuario y “pass” como contraseña. 3. Pulsar el botón “Entrar”.
Verificación	– Aparece un mensaje indicando que los datos de usuario no son correctos para acceder a la aplicación.

Identificador	PR-M-4
Módulo	Modificar datos de usuario
Casos de uso	CU-M-2
Precondiciones	– El usuario debe haberse autenticado en la aplicación móvil ColDes.
Procedimiento	<ol style="list-style-type: none"> 1. Acceder a la zona de datos de usuario a partir del botón de la barra de acción de la aplicación. 2. Modificar los datos de usuario. 3. Pulsar sobre el botón “Actualizar”.
Verificación	<ul style="list-style-type: none"> – Se muestra un mensaje de verificación indicando que la operación se ha realizado con éxito. – Se actualiza el registro de la tabla “user” de la base de datos correspondiente al usuario.

Identificador	PR-M-5
Módulo	Modificar datos de usuario
Casos de uso	CU-M-2
Precondiciones	– El usuario debe haberse autenticado en la aplicación móvil ColDes.
Procedimiento	<ol style="list-style-type: none"> 1. Acceder a la zona de datos de usuario a partir del botón de la barra de acción de la aplicación. 2. Pulsar sobre el <i>check box</i> que permite el cambio de contraseña. 3. Introducir distintas contraseñas en la nueva contraseña y el campo de confirmación de contraseña. 4. Pulsar sobre el botón “Actualizar”.
Verificación	<ul style="list-style-type: none"> – Se muestra un mensaje de error indicando que ambas contraseñas deben ser iguales. – No se realizan cambios en la base de datos.

Identificador	PR-M-6
Módulo	Añadir recursos
Casos de uso	CU-M-3
Precondiciones	– El usuario debe haberse autenticado en la aplicación móvil ColDes.
Procedimiento	<ol style="list-style-type: none"> 1. Acceder a la pestaña de recursos de la aplicación móvil. 2. Pulsar sobre el botón con forma de clip que permite añadir recursos a la cuenta. 3. Escoger la opción de añadir un recurso desde la galería. 4. Escoger un recurso con extensión <i>.jpg</i>, <i>.png</i>, <i>.avi</i>, <i>.mp4</i>, <i>.flv</i> o <i>.mp3</i>. 5. Introducir el nombre del recurso, el comentario y las etiquetas asociadas al recurso. 6. Pulsar sobre el botón “Aceptar”. 7. Refrescar la lista de recursos pulsando sobre el botón de la zona superior de la pantalla.
Verificación	<ul style="list-style-type: none"> – Se muestra un mensaje de verificación indicando que la operación se ha realizado con éxito. – El nuevo recurso se añade a la lista de recursos del usuario. – Se crea un nuevo registro en la tabla “userresource” de la base de datos con los datos del recurso añadido. – Se añade un nuevo archivo correspondiente al recurso seleccionado en el sistema de ficheros.

Identificador	PR-M-7
Módulo	Añadir recursos
Casos de uso	CU-M-3
Precondiciones	– El usuario debe haberse autenticado en la aplicación móvil ColDes.
Procedimiento	<ol style="list-style-type: none"> 1. Acceder a la pestaña de recursos de la aplicación móvil. 2. Pulsar sobre el botón con forma de clip que permite añadir recursos a la cuenta. 3. Escoger la opción de añadir un recurso desde la galería. 4. Escoger un recurso con una extensión que no se corresponda a las siguientes: <i>.jpg</i>, <i>.png</i>, <i>.avi</i>, <i>.mp4</i>, <i>.flv</i> o <i>.mp3</i>.
Verificación	– Se muestra un mensaje de error indicando que el fichero no es soportado por la aplicación.

Identificador	PR-M-8
Módulo	Añadir recursos
Casos de uso	CU-M-3
Precondiciones	– El usuario debe haberse autenticado en la aplicación móvil ColDes.
Procedimiento	<ol style="list-style-type: none"> 1. Acceder a la pestaña de recursos de la aplicación móvil. 2. Pulsar sobre el botón con forma de clip que permite añadir recursos a la cuenta. 3. Escoger la opción de tomar una foto para añadirla como recurso. 4. Tomar la foto. 5. Introducir el nombre del recurso, el comentario y las etiquetas asociadas al recurso. 6. Pulsar sobre el botón “Aceptar”. 7. Refrescar la lista de recursos pulsando sobre el botón de la zona superior de la pantalla.
Verificación	<ul style="list-style-type: none"> – Se muestra un mensaje de verificación indicando que la operación se ha realizado con éxito. – El nuevo recurso se añade a la lista de recursos del usuario. – Se crea un nuevo registro en la tabla “userresource” de la base de datos con los datos del recurso añadido. – Se crea un nuevo archivo de imagen en el sistema de ficheros con el contenido de la foto tomada.

Identificador	PR-M-9
Módulo	Añadir recursos
Casos de uso	CU-M-3
Precondiciones	– El usuario debe haberse autenticado en la aplicación móvil ColDes.
Procedimiento	<ol style="list-style-type: none"> 1. Acceder a la pestaña de recursos de la aplicación móvil. 2. Pulsar sobre el botón con forma de clip que permite añadir recursos a la cuenta. 3. Escoger la opción de tomar un vídeo para añadirlo como recurso. 4. Tomar el vídeo. 5. Introducir el nombre del recurso, el comentario y las etiquetas asociadas al recurso. 6. Pulsar sobre el botón “Aceptar”. 7. Refrescar la lista de recursos pulsando sobre el botón de la zona superior de la pantalla.
Verificación	– Se muestra un mensaje de verificación indicando que la operación se

	<p>ha realizado con éxito.</p> <ul style="list-style-type: none"> – El nuevo recurso se añade a la lista de recursos del usuario. – Se crea un nuevo registro en la tabla “userresource” de la base de datos con los datos del recurso añadido. – Se crea un nuevo archivo de vídeo en el sistema de ficheros con el contenido de vídeo capturado.
--	---

Identificador	PR-M-10
Módulo	Añadir recursos
Casos de uso	CU-M-3
Precondiciones	<ul style="list-style-type: none"> – El usuario debe haberse autenticado en la aplicación móvil ColDes.
Procedimiento	<ol style="list-style-type: none"> 1. Acceder a la pestaña de recursos de la aplicación móvil. 2. Pulsar sobre el botón con forma de clip que permite añadir recursos a la cuenta. 3. Escoger la opción de añadir un recurso de texto. 4. Introducir el nombre del recurso, el contenido textual, el comentario y las etiquetas asociadas al recurso. 5. Pulsar sobre el botón “Aceptar”. 6. Refrescar la lista de recursos pulsando sobre el botón de la zona superior de la pantalla.
Verificación	<ul style="list-style-type: none"> – Se muestra un mensaje de verificación indicando que la operación se ha realizado con éxito. – El nuevo recurso se añade a la lista de recursos del usuario. – Se crea un nuevo registro en la tabla “userresource” de la base de datos con los datos del recurso añadido.

Identificador	PR-M-11
Módulo	Visualizar recursos
Casos de uso	CU-M-4
Precondiciones	<ul style="list-style-type: none"> – El usuario debe haberse autenticado en la aplicación móvil ColDes. – PR-M-6, PR-M-8, PR-M-9 o PR-M-10
Procedimiento	<ol style="list-style-type: none"> 1. Acceder a la pestaña de recursos de la aplicación móvil. 2. Pulsar sobre cualquiera de los recursos de la lista de recursos.
Verificación	<ul style="list-style-type: none"> – Se abre una nueva pantalla en la que se muestra el contenido del recurso. Si el recurso no ha sido etiquetado, se muestra un mensaje indicando que no hay etiquetas asociadas al recurso.

Identificador	PR-M-12
Módulo	Visualizar recursos
Casos de uso	CU-M-4
Precondiciones	– El usuario debe haberse autenticado en la aplicación móvil ColDes.
Procedimiento	<ol style="list-style-type: none"> 1. Acceder a la pestaña de recursos de la aplicación móvil. 2. La lista de recursos está vacía.
Verificación	– Aparece un mensaje indicando que no existen recursos asociados a la cuenta de usuario.

Identificador	PR-M-13
Módulo	Eliminar recursos
Casos de uso	CU-M-5
Precondiciones	<ul style="list-style-type: none"> – El usuario debe haberse autenticado en la aplicación móvil ColDes. – PR-M-6, PR-M-8, PR-M-9 o PR-M-10
Procedimiento	<ol style="list-style-type: none"> 1. Acceder a la pestaña de recursos de la aplicación móvil. 2. Pulsar sobre cualquiera de los recursos de la lista de recursos. 3. Pulsar sobre el botón con forma de papelera que permite eliminar un recurso. 4. Aceptar el mensaje de confirmación. 5. Refrescar la lista de recursos pulsando sobre el botón de la zona superior de la pantalla.
Verificación	<ul style="list-style-type: none"> – Se muestra un mensaje de verificación indicando que la operación se ha realizado con éxito. – El recurso seleccionado se elimina de la lista de recursos del usuario. – Se elimina el registro de la tabla “userresource” de la base de datos asociado al recurso, así como los registros de las tablas “tagresource”, “itemresource” y “roomresource” asociados al recurso eliminado. – Se elimina el archivo del sistema de ficheros asociado al recurso.

Identificador	PR-M-14
Módulo	Eliminar recursos
Casos de uso	CU-M-5
Precondiciones	<ul style="list-style-type: none"> – El usuario debe haberse autenticado en la aplicación móvil ColDes. – PR-M-6, PR-M-8, PR-M-9 o PR-M-10
Procedimiento	<ol style="list-style-type: none"> 1. Acceder a la pestaña de recursos de la aplicación móvil. 2. Pulsar sobre cualquiera de los recursos de la lista de recursos.

	<ol style="list-style-type: none"> 3. Pulsar sobre el botón con forma de papelera que permite eliminar un recurso. 4. Cancelar el mensaje de confirmación. 5. Refrescar la lista de recursos pulsando sobre el botón de la zona superior de la pantalla.
Verificación	<ul style="list-style-type: none"> – El recurso seleccionado permanece la lista de recursos del usuario. – No se realizan cambios en la base de datos. – No se realizan cambios en el sistema de ficheros.

Identificador	PR-M-15
Módulo	Añadir etiquetas
Casos de uso	CU-M-6
Precondiciones	<ul style="list-style-type: none"> – El usuario debe haberse autenticado en la aplicación móvil ColDes.
Procedimiento	<ol style="list-style-type: none"> 1. Pulsar sobre el botón de acciones de la barra de acción de la aplicación. 2. Acceder a la zona de etiquetas. 3. Pulsar sobre el botón que permite añadir etiquetas a la cuenta. 4. Introducir el nombre de la etiqueta y el color asociado a la misma. 5. Pulsar sobre el botón “Aceptar”. 6. Refrescar la lista de etiquetas pulsando sobre el botón de la zona superior de la pantalla.
Verificación	<ul style="list-style-type: none"> – Se muestra un mensaje de verificación indicando que la operación se ha realizado con éxito. – La nueva etiqueta se añade a la lista de recursos del usuario. – Se crea un nuevo registro en la tabla “usertag” de la base de datos con los datos de la etiqueta añadida.

Identificador	PR-M-16
Módulo	Añadir etiquetas
Casos de uso	CU-M-6
Precondiciones	<ul style="list-style-type: none"> – El usuario debe haberse autenticado en la aplicación móvil ColDes.
Procedimiento	<ol style="list-style-type: none"> 1. Pulsar sobre el botón de acciones de la barra de acción de la aplicación. 2. Acceder a la zona de etiquetas. 3. Pulsar sobre el botón que permite añadir etiquetas a la cuenta. 4. Introducir el nombre de la etiqueta y el color asociado a la misma. 5. Pulsar sobre el botón “Cancelar”.

	6. Refrescar la lista de etiquetas pulsando sobre el botón de la zona superior de la pantalla.
Verificación	<ul style="list-style-type: none"> – No se realizan cambios en la lista de recursos del usuario. – No se realizan cambios en la base de datos.

Identificador	PR-M-17
Módulo	Etiquetar recurso
Casos de uso	CU-M-7
Precondiciones	<ul style="list-style-type: none"> – El usuario debe haberse autenticado en la aplicación móvil ColDes. – PR-M-15
Procedimiento	<ol style="list-style-type: none"> 1. Acceder a la pestaña de recursos de la aplicación móvil. 2. Pulsar sobre cualquiera de los recursos de la lista de recursos. 3. Pulsar sobre el botón que permite etiquetar el recurso. 4. Elegir la etiqueta a asociar al recurso. 5. Pulsar sobre el botón “Aceptar”. 6. Refrescar la lista de etiquetas del recurso.
Verificación	<ul style="list-style-type: none"> – Se muestra un mensaje de verificación indicando que la operación se ha realizado con éxito. – La etiqueta seleccionada se añade a la lista de etiquetas del recurso. – Se crea un nuevo registro en la tabla “tagresource” de la base de datos que relaciona la etiqueta con el recurso seleccionado.

Identificador	PR-M-18
Módulo	Etiquetar recurso
Casos de uso	CU-M-7
Precondiciones	<ul style="list-style-type: none"> – El usuario debe haberse autenticado en la aplicación móvil ColDes. – PR-M-15
Procedimiento	<ol style="list-style-type: none"> 1. Acceder a la pestaña de recursos de la aplicación móvil. 2. Pulsar sobre cualquiera de los recursos de la lista de recursos. 3. Pulsar sobre el botón que permite etiquetar el recurso. 4. Elegir la etiqueta a asociar al recurso. 5. Pulsar sobre el botón “Cancelar”. 6. Refrescar la lista de etiquetas del recurso.
Verificación	<ul style="list-style-type: none"> – No se realizan cambios en la lista de etiquetas del recurso. – No se realizan cambios en la base de datos.

Identificador	PR-M-19
Módulo	Etiquetar recurso
Casos de uso	CU-M-7
Precondiciones	– El usuario debe haberse autenticado en la aplicación móvil ColDes.
Procedimiento	<ol style="list-style-type: none"> 1. Acceder a la pestaña de recursos de la aplicación móvil. 2. Pulsar sobre cualquiera de los recursos de la lista de recursos. 3. Pulsar sobre el botón que permite etiquetar el recurso. 4. La lista de etiquetas está vacía.
Verificación	– Aparece un mensaje indicando que no existen etiquetas asociadas a la cuenta de usuario.

Identificador	PR-M-20
Módulo	Visualizar lista de recursos de una etiqueta
Casos de uso	CU-M-8
Precondiciones	<ul style="list-style-type: none"> – El usuario debe haberse autenticado en la aplicación móvil ColDes. – PR-M-15 – PR-M-6, PR-M-8, PR-M-9 o PR-M-10 – PR-M-17
Procedimiento	<ol style="list-style-type: none"> 1. Pulsar sobre el botón de acciones de la barra de acción de la aplicación. 2. Acceder a la zona de etiquetas. 3. Seleccionar una de las etiquetas de la lista.
Verificación	– Se abre una nueva pantalla en la que se muestran los recursos asociados a la etiqueta.

Identificador	PR-M-21
Módulo	Visualizar lista de recursos de una etiqueta
Casos de uso	CU-M-8
Precondiciones	<ul style="list-style-type: none"> – El usuario debe haberse autenticado en la aplicación móvil ColDes. – PR-M-15 – PR-M-6, PR-M-8, PR-M-9 o PR-M-10 – PR-M-17
Procedimiento	<ol style="list-style-type: none"> 4. Pulsar sobre el botón de acciones de la barra de acción de la aplicación. 5. Acceder a la zona de etiquetas. 6. Seleccionar una de las etiquetas de la lista.

	7. La lista de recursos etiquetados por la etiqueta seleccionada está vacía.
Verificación	<ul style="list-style-type: none"> – Se abre una nueva pantalla en la que se muestran los recursos asociados a la etiqueta. – Se muestra un mensaje indicando que no existen recursos asociados a la etiqueta.

Identificador	PR-M-22
Módulo	Eliminar etiquetas
Casos de uso	CU-M-9
Precondiciones	<ul style="list-style-type: none"> – El usuario debe haberse autenticado en la aplicación móvil ColDes. – PR-M-15
Procedimiento	<ol style="list-style-type: none"> 1. Pulsar sobre el botón de acciones de la barra de acción de la aplicación. 2. Acceder a la zona de etiquetas. 3. Seleccionar una de las etiquetas de la lista. 4. Pulsar sobre el botón con forma de papelera que permite eliminar una etiqueta. 5. Aceptar el mensaje de confirmación. 6. Refrescar la lista de etiquetas pulsando sobre el botón de la zona superior de la pantalla.
Verificación	<ul style="list-style-type: none"> – Se muestra un mensaje de verificación indicando que la operación se ha realizado con éxito. – La etiqueta seleccionada se elimina de la lista de etiquetas del usuario. – Se eliminan los registros de la tabla “usertag” y “tagresource” de la base de datos relacionados con la etiqueta eliminada.

Identificador	PR-M-23
Módulo	Eliminar etiquetas
Casos de uso	CU-M-9
Precondiciones	<ul style="list-style-type: none"> – El usuario debe haberse autenticado en la aplicación móvil ColDes. – PR-M-15
Procedimiento	<ol style="list-style-type: none"> 1. Pulsar sobre el botón de acciones de la barra de acción de la aplicación. 2. Acceder a la zona de etiquetas. 3. Seleccionar una de las etiquetas de la lista. 4. Pulsar sobre el botón con forma de papelera que permite eliminar una

	<p>etiqueta.</p> <p>5. Cancelar el mensaje de confirmación.</p> <p>6. Refrescar la lista de etiquetas pulsando sobre el botón de la zona superior de la pantalla.</p>
Verificación	<ul style="list-style-type: none"> – La etiqueta seleccionada permanece en la lista de etiquetas del usuario. – No se realizan cambios en la base de datos.

5.2 Análisis de resultados

En la presente sección se analizan los resultados obtenidos de las pruebas de evaluación descritas en la sección anterior. Concretamente, se recogen los resultados obtenidos de aplicar los casos de prueba del apartado *Casos de prueba*, para después extraer las conclusiones obtenidas del proceso de evaluación.

En la *Tabla 23* se presentan los resultados obtenidos del proceso de evaluación. Para cada prueba asociada a cada aplicación se muestra el número de tentativa, la fecha de realización, el resultado obtenido del caso de prueba y una descripción del proceso de evaluación para el caso de prueba concreto, en caso de ser necesario.

	Prueba	Fecha	# Tentativa	Resultado	Descripción
Aplicación web	PR-W-1	27/05/2013	1	Satisfactorio	
	PR-W-2	27/05/2013	1	Satisfactorio	
	PR-W-3	27/05/2013	1	Satisfactorio	
	PR-W-4	27/05/2013	1	Satisfactorio	
	PR-W-5	27/05/2013	1	Satisfactorio	
	PR-W-6	27/05/2013	1	Satisfactorio	
	PR-W-7	27/05/2013	1	Satisfactorio	
	PR-W-8	27/05/2013	1	Satisfactorio	
	PR-W-9	27/05/2013	1	Satisfactorio	
	PR-W-10	27/05/2013	1	Satisfactorio	
	PR-W-11	27/05/2013	1	Satisfactorio	
	PR-W-12	27/05/2013	1	Satisfactorio	
	PR-W-13	27/05/2013	1	Satisfactorio	
	PR-W-14	27/05/2013	1	Satisfactorio	
	PR-W-15	27/05/2013	1	Satisfactorio	
	PR-W-16	27/05/2013	1	Satisfactorio	

	PR-W-17	27/05/2013	1	Satisfactorio	
	PR-W-18	27/05/2013	1	Satisfactorio	
	PR-W-19	27/05/2013	1	Satisfactorio	
	PR-W-20	27/05/2013	1	Satisfactorio	
	PR-W-21	27/05/2013	1	Satisfactorio	
	PR-W-22	27/05/2013	1	Satisfactorio	
	PR-W-23	27/05/2013	1	Satisfactorio	
Aplicación móvil	PR-M-1	28/05/2013	1	Satisfactorio	
	PR-M-2	28/05/2013	1	Satisfactorio	
	PR-M-3	28/05/2013	1	Satisfactorio	
	PR-M-4	28/05/2013	1	Satisfactorio	
	PR-M-5	28/05/2013	1	Satisfactorio	
	PR-M-6	28/05/2013	1	Satisfactorio	
	PR-M-7	28/05/2013	1	Satisfactorio	
	PR-M-8	28/05/2013	1	Satisfactorio	
	PR-M-9	28/05/2013	1	Satisfactorio	
	PR-M-10	28/05/2013	1	Satisfactorio	
	PR-M-11	28/05/2013	1	Satisfactorio	
	PR-M-12	28/05/2013	1	Satisfactorio	
	PR-M-13	28/05/2013	1	Satisfactorio	
	PR-M-14	28/05/2013	1	Satisfactorio	
	PR-M-15	28/05/2013	1	Satisfactorio	
	PR-M-16	28/05/2013	1	Satisfactorio	
	PR-M-17	28/05/2013	1	Satisfactorio	
	PR-M-18	28/05/2013	1	Satisfactorio	
	PR-M-19	28/05/2013	1	Satisfactorio	
	PR-M-20	28/05/2013	1	Satisfactorio	
	PR-M-21	28/05/2013	1	Satisfactorio	

Tabla 23: Resultados del proceso de evaluación.

Como se puede observar, todos los casos de uso son verificados en la primera tentativa del proceso de evaluación, por lo que el sistema queda verificado.

6 Conclusions

This chapter summarizes the main contributions of this project, lists the problems encountered along its execution and establishes recommendations for future work. To conclude, different personal opinions related with the performance of this work are presented.

6.1 Contributions

The main contribution of this project is the development of a computer-based collaborative environment for supporting the co-design process in a distributed way. Accordingly, this project has extended the basic functionality of the design tool ColDes, allowing the participants to create rich affinity diagrams. It is necessary to highlight that these affinity diagrams are persistent. In this way, one participant can present their ideas at different times and recover this affinity diagram when it is needed.

Similarly, an Android mobile application has been developed for allowing participants to complete the ideas presented in the information items of the affinity diagram. In particular, through this mobile application, the participants can collect new multimedia resources and include them in the affinity diagram. To do this, they just need to load existing affinity diagrams, which preserves the changes made by other users in the workspace. In this way, it is easier to both understand and share the ideas with people who are in different places at different times.

6.2 Future work

This section lists some future works that may arise from the realization of this work.

The first next steps should be oriented to develop the mobile version of ColDes. In the current situation, participants need to access to a computer to make changes in the affinity diagrams or to see the changes made by other participants. By having this mobile application, participants not only would be allowed to collect multimedia resources in a contextual manner, but also to add his/her ideas at the same time in which a resource is captured.

Second steps would be oriented to merging the group sketching technique with the affinity diagram workspace in order to take advantage of the best of both co-design techniques. On the one hand, participants could add a new type of resource corresponding to the designs of a group sketching workspace. In this way, the participants could express their ideas through drawings and initial schemes, which assist understanding of the ideas from the other users, facilitating the process of making an affinity diagram. On the other hand, by using group sketching tools, users could create initial schemes of the ideas they have thought to solve the problem that arises in the affinity diagram. In other words, this would be the first step before creating the affinity diagram.

More further steps would be directed to convert ColDes in a multitouch application. In this way, all aspects of groupware matrix would be covered (see [Figura 1](#)). In the current application, multiple users can simultaneously participate in creating an affinity diagram, but several users can't create information items at the same time. With multitouch, capabilities it would be possible to collect events from several activities within a workspace, and multiple participants could participate at the same time.

6.3 Problems encountered

On the development of the web application, main problems arisen because of the technology that has been used. Flash is a multimedia platform launched in 1996 that, by the year 2000, had become the standard for video playback and interactive multimedia web pages. It became a standard tool that could be used in multiplatform browsers. However, from that moment, the presence of Flash on the web has decreased. The amount of documentation that can be found on website has decreased dramatically. The pages dedicated to show practical examples of using Flex and ActionScript are now obsolete, and is harder to find documentation to specific problems. As Steve Jobs published in his open letter entitled "Thoughts on Flash" , *"Flash is no longer necessary to watch video or consume any kind of web content"* and that *"the new open standards created in the mobile era, such as HTML5, will win on mobile devices (and PCs too)."* [31].

In the context of this project, this lack of documentation affected to the use of modal boxes that were used for implementing information items. Furthermore, once an appropriate component was found, it was very difficult to find information on how to restrict the movement of these components on other elements in the stage. Therefore, many functions were implemented from scratch.

Regarding to the development of the mobile application, problems encountered were related with communication issues with the database. Although Android is a *booming* technology and it is possible to find lots of documentation on communication with databases, there are no specific solutions that would allow the communication with the database safely through both applications. Therefore, it was necessary to create a Web service that allowed the communication of the database through any technology. This meant an extra work that had not been initially estimated.

6.4 Personal opinions

The realization of this final project has supposed a great challenge. Particularly, and as it has been mentioned in the previous section, there have arisen several development problems, which have hinder its completion. However, the project planning has been achieved on time and different new skills and abilities have been obtained through its realization. To conclude this chapter, some of these abilities are described below. Previous skills gained from my degree and applied to this project are also listed.

Firstly, I would like to highlight different skills, which have been learned throughout my degree, applied in this project. For example, a data model that allows data persistence has been designed. Its design has been based on my databases design skills acquired from practical work in laboratories during my degree. Similarly, some theoretical concepts have been reinforced. This is the case of concepts such as collaboration, distributed system or process model.

To conclude, several new concepts and abilities learned from the realization of this project have been gained. One of these abilities is the development of a *middleware*, which allows communication between different systems independently of the technology used. To develop this *middleware*, it has been applied a new programming concept for me, design patterns, which provide a better way to structure the code of the project. Their use has supposed great advantages. In particular, it allows externalizing information from the system functionality. Another ability learned has been the application of design guidelines for the user interface. Specifically, the design guidelines of Android have been used to create a usable and intuitive user interface for the mobile application.

7 Bibliografía

- [1] J. M. B. García, "ColDes: Desarrollo de una herramienta web para el dibujo gráfico colaborativo," [Online]. Available: <http://e-archivo.uc3m.es/bitstream/10016/14787/1/PFC-ColDes-JoseMiguelBlancoGarcia.pdf>. [Accessed 20 abril 2013].
- [2] P. J. S. Elizabeth B.-N. Sanders, "Co-creation and the new landscapes of design," *CoDesign: International Journal of CoCreation in Design and the Arts*, vol. IV, no. 1, pp. 5-18, 2008.
- [3] C. Dictionary, "Definition of "collaboration"," [Online]. Available: <http://www.collinsdictionary.com/dictionary/english/collaboration>. [Accessed 08 junio 2013].
- [4] J. F. G. A. R. B. Alan Dix, "Groupware," in *Human-Computer Interaction*, Prentice Hall, 2004.
- [5] "Service Design Tools: Communication methods supporting design processes," 2008. [Online]. Available: <http://www.servicedesigntools.org/taxonomy/term/1>. [Accessed 3 abril 2013].
- [6] S. Greenberg, M. Roseman and D. Webster, *GroupSketch*, Department of Computer Science, University of Calgary, Calgary, Canada.
- [7] "MindTools: Brainstorming," [Online]. Available: <http://www.mindtools.com/brainstm.html>. [Accessed 09 abril 2013].
- [8] C. T. Nicola Morelli, *New Representation Techniques for Designing in a Systemic Perspective*, Stockholm, 2007.
- [9] "Ley de Presupuestos Generales del Estado para el año 2013," 27 diciembre 2012. [Online]. Available: http://noticias.juridicas.com/base_datos/Fiscal/I17-2012.html. [Accessed 06 abril 2013].
- [10] "1&1: Alquiler de un VPS Linux," [Online]. Available: <http://www.1and1.es/VirtualServerXL>. [Accessed 6 abril 2013].
- [11] "International Standard ISO/IEC 12207. Software Life Cycle Processes," [Online]. Available: <http://www.abelia.com/docs/12207cpt.pdf>. [Accessed 20 abril 2013].
- [12] "Desarrollo de Software Iterativo e Incremental," 14 febrero 2007. [Online]. Available: <http://fernandosoriano.com.ar/?p=14>. [Accessed 27 abril 2013].
- [13] R. S. M. M. Armando Cabrera, "Procesos de Ingeniería del Software," [Online]. Available: <http://www.slideshare.net/rfsolano/procesos-de-ingenieria-del-software>. [Accessed 27 abril 2013].
- [14] Volere, "Plantilla de Especificación de Requisitos," febrero 2006. [Online]. Available: http://www.volere.co.uk/pdf%20files/template_es.pdf. [Accessed 13 abril 2013].
- [15] D. Gelperin, "Methods & Tools," [Online]. Available: <http://www.methodsandtools.com/archive/archive.php?id=8>. [Accessed 25 mayo 2013].

- [16] E. G. Daryl Kulak, "Use Cases: Requirements in Context," *Software Engineering Notes*, vol. 26, no. 1, p. 101, 2001.
- [17] R. H. R. J. J. V. Erich Gamma, *Design Patterns: Elements of Reusable Object-Oriented Software*, Addison-Wesley Professional, 1994.
- [18] "List of web service protocols," 23 marzo 2013. [Online]. Available: http://en.wikipedia.org/wiki/List_of_web_service_protocols. [Accessed 09 mayo 2013].
- [19] "W3C: Simple Object Access Protocol (SOAP) 1.1," 08 mayo 2000. [Online]. Available: <http://www.w3.org/TR/2000/NOTE-SOAP-20000508/>. [Accessed 09 mayo 2013].
- [20] D. M. Elkstein, "Learn REST," 09 febrero 2008. [Online]. Available: <http://rest.elkstein.org/>. [Accessed 09 mayo 2013].
- [21] J. C. D. M. Deepak Alur, *Core J2EE™ Patterns: Best Practices and Design Strategies*, Second Edition, Prentice Hall, 2003.
- [22] "msdn: Enterprise Solution Patterns," [Online]. Available: <http://msdn.microsoft.com/en-us/library/ms978717.aspx>. [Accessed 11 mayo 2013].
- [23] M. Dostál, "User Acceptance of the Microsoft Ribbon User Interface," *DNCOCO'10 Proceedings of the 9th WSEAS international conference on Data networks, communications, computers*, pp. 143 - 149, 2010.
- [24] "Eclipse," [Online]. Available: <http://www.eclipse.org/>. [Accessed 18 mayo 2013].
- [25] "Adobe Flex Builder 3.0.2 Professional Eclipse Plug-in," [Online]. Available: https://www.adobe.com/cfusion/tdrc/index.cfm?product=flex_eclipse. [Accessed 18 mayo 2013].
- [26] "Adobe Flash Builder 4.7 Premium," [Online]. Available: <http://www.adobe.com/es/products/flash-builder.html>. [Accessed 18 mayo 2013].
- [27] "Developers Android: ADT Plugin," [Online]. Available: <http://developer.android.com/tools/sdk/eclipse-adt.html>. [Accessed 18 mayo 2013].
- [28] "NetBeans IDE," [Online]. Available: <https://netbeans.org/>. [Accessed 18 mayo 2013].
- [29] "Developers Android: Supporting Multiple Screens," [Online]. Available: http://developer.android.com/guide/practices/screens_support.html. [Accessed 18 mayo 2013].
- [30] "Sample Test Case Template with Examples," [Online]. Available: <http://www.softwaretestinghelp.com/test-case-template-examples/>. [Accessed 27 mayo 2013].
- [31] S. Jobs, "Thoughts on Flash," abril 2010. [Online]. Available: <http://www.apple.com/hotnews/thoughts-on-flash/>. [Accessed 09 junio 2013].
- [32] "Google App Inventor," [Online]. Available: <http://www.appinventor.es/configuracion.php>. [Accessed 9 marzo 2013].

- [33] "Open Handset Alliance," [Online]. Available: <http://www.openhandsetalliance.com/>. [Accessed 9 marzo 2013].
- [34] "Android SDK," [Online]. Available: <http://developer.android.com/sdk/index.html>. [Accessed 09 marzo 2013].
- [35] "Sociable.co: Google Play will hit one million apps this June," [Online]. Available: <http://sociable.co/mobile/google-play-will-hit-one-billion-apps-this-june/>. [Accessed 9 marzo 2013].
- [36] "Statista: Google überholt Apple," [Online]. Available: <http://de.statista.com/themen/882/apps-app-stores/infografik/810/anzahl-der-verfuegbaren-apps-in-den-top-app-stores/>.
- [37] "RTVE: Android supera los 500 millones de dispositivos activados a nivel global," [Online]. Available: <http://www.rtve.es/noticias/20120912/android-supera-500-millones-dispositivos-activados-nivel-global/562914.shtml>. [Accessed 9 marzo 2013].
- [38] "Apple developer: iOS Developer Program," [Online]. Available: <https://developer.apple.com/programs/ios/>. [Accessed 9 marzo 2013].
- [39] "Apple: Estadísticas de ventas," [Online]. Available: <http://www.infobae.com/notas/677656-Apple-100-millones-de-iPad-vendidos-en-dos-anos-y-medio.html>. [Accessed 9 marzo 2013].
- [40] "Google Play: Developer console," [Online]. Available: <https://play.google.com/apps/publish/v2/signup/?pli=1>. [Accessed 9 marzo 2013].
- [41] E. Tan, "Comparativa Windows Phone vs Android vs iPhone," [Online]. Available: <http://myphonedeads.co.uk/blog/33-the-smartphone-os-complete-comparison-chart>. [Accessed 9 marzo 2013].
- [42] "msdn Microsoft: Desarrollo de aplicaciones móviles," [Online]. Available: <http://msdn.microsoft.com/es-es/ff380145>. [Accessed 9 marzo 2013].
- [43] Techerald.com, "JavaFX facilidad de uso y el futuro," 8 Agosto 2011. [Online]. Available: <http://techerald.com/page/javafx-facilidad-de-uso-y-el-futuro.html>. [Accessed 1 febrero 2012].
- [44] E. V. Hippel, Democratizing innovation: The evolving phenomenon of user innovation, 2005.
- [45] "Android Developers: Navigation Drawer," [Online]. Available: <http://developer.android.com/design/patterns/navigation-drawer.html>. [Accessed 09 junio 2013].

Anexo I. Control de versiones

Versión	Fecha de finalización	Descripción
0.1	09 mar. 2013	Redacción de sección: 2.2 Revisión de tecnologías móviles
0.2	15 mar. 2013	Redacción de secciones: 2.1 Revisión teórica de técnicas de co-diseño
0.7	20 mar. 2013	Redacción de sección: 3.1 Estimación de recursos
0.9	23 mar. 2013	Redacción de secciones: 3.2 Presupuesto
1.0	25 mar. 2013	Revisión de capítulos: 2 Estado de la cuestión 3 Gestión de proyecto software
1.2	2 abr. 2013	Redacción de la sección: 4.1 Descripción de la solución 4.2 El proceso de desarrollo Modelo de proceso
1.5	6 abr. 2013	Redacción de la sección: 4.2 El proceso de desarrollo Análisis Definición de requisitos Anexo III. Especificación de requisitos
1.8	11 abr. 2013	Redacción de la sección: 4.2 El proceso de desarrollo Análisis Casos de uso
2.0	13 abr. 2013	Revisión de la sección: 4.2 El proceso de desarrollo Análisis
2.2	16 abr. 2013	Redacción de la sección: 4.2 El proceso de desarrollo Diseño Diseño de sistema
2.5	22 abr. 2013	Redacción de la sección: 4.2 El proceso de desarrollo Diseño Diseño detallado

		Capa de persistencia
2.8	26 abr. 2013	Redacción de la sección: 4.2 El proceso de desarrollo Diseño Diseño detallado Interfaz
2.9	29 abr. 2013	Redacción del capítulo: Anexo IV. Prototipo
3.0	3 may. 2013	Revisión de la sección: 4.2 El proceso de desarrollo Diseño
3.2	5 may. 2013	Redacción de la sección 4.2 El proceso de desarrollo Implementación Entorno de desarrollo
4.0	12 may. 2013	Redacción de la sección 4.2 El proceso de desarrollo Implementación Organización del proyecto
4.5	24 may. 2013	Redacción de la sección 5.1 Proceso de evaluación
5.0	28 may. 2013	Redacción de la sección 5.2 Análisis de resultados
5.5	4 jun. 2013	Redacción del capítulo: 1 Introduction
5.8	8 jun. 2013	Redacción del capítulo: Anexo II: Seguimiento del trabajo fin de grado
6.0	11 jun. 2013	Redacción del capítulo: 6 Conclusions

Tabla 24: Control de versiones.

Anexo II. Seguimiento del trabajo fin de grado

En este capítulo se detalla la forma de seguimiento del trabajo fin de grado. Para ello, en primer lugar, se establece la forma en que el trabajo ha sido desarrollado. A continuación, se presenta la planificación realizada al comienzo del trabajo de fin de grado, explicando las tareas establecidas y los periodos de tiempo asociados a la realización de cada una de ellas.

Forma de seguimiento

La forma de seguimiento del trabajo fin de grado se ha basado en una reunión semanal con el tutor del trabajo de fin de grado. Estas reuniones se ha realizado todas las semanas de los meses de abril y mayo, meses en los que se ha escrito el grueso de este documento. En estas reuniones se comprobaba el estado de la memoria y se establecían hitos a cumplir para la siguiente reunión.

En todos los meses de duración del trabajo, se han realizado reuniones esporádicas para comparar el estado de la implementación con respecto a la planificación inicial.

Planificación del trabajo fin de grado

En la *Figura 36* se muestra un diagrama de Gantt con las tareas planificadas para la implementación de la aplicación móvil de ColDes y la adaptación de la aplicación web para introducir nuevos *workspaces* para la creación de diagramas de afinidad. A cada una de esas tareas se le asocia un periodo de tiempo estimado en el que se planifica que debe ser concluida.

Id.	Nombre de tarea	Comienzo	Fin	Duración (días)	ene 2013				feb 2013				mar 2013				abr 2013				may 2013			
					6/1	13/1	20/1	27/1	3/2	10/2	17/2	24/2	3/3	10/3	17/3	24/3	31/3	7/4	14/4	21/4	28/4	5/5	12/5	19/5
1	M: Workspaces	01/01/2013	18/01/2013	14d																				
2	M: Recursos	21/01/2013	15/02/2013	20d																				
3	M: Etiquetas	18/02/2013	01/03/2013	10d																				
4	M: Modelo de datos	04/03/2013	22/03/2013	15d																				
5	W: Multi-tool	25/03/2013	29/03/2013	5d																				
6	W: Diagrama de afinidad	01/04/2013	26/04/2013	20d																				
7	W: Recursos multimedia	29/04/2013	10/05/2013	10d																				
8	W: Modelo de datos	13/05/2013	31/05/2013	15d																				

Figura 36: Diagrama de Gantt: Planificación inicial del trabajo fin de grado.

Las tareas están precedidas en las consonantes M o W, según traten de la aplicación móvil o de la aplicación web, respectivamente. A continuación, se describe en qué consiste cada una de las tareas del diagrama de Gantt.

- 1) **Aplicación móvil – Workspaces:** Implementación del módulo de *workspaces* de la aplicación móvil. Incluye las operaciones de creación de *workspaces* y visualización de la lista de *workspaces* de un usuario.
- 2) **Aplicación móvil – Recursos:** Implementación del módulo de recursos de la aplicación móvil. Se trata del módulo principal de la aplicación móvil, que incluye la visualización de la lista de recursos de un usuario, la inclusión de nuevos recursos a la cuenta de usuario, tanto a partir de la galería del dispositivo como a partir de la cámara del mismo, y la eliminación de recursos.
- 3) **Aplicación móvil – Etiquetas:** Implementación del módulo de etiquetas de la aplicación móvil. Incluye las operaciones de creación y eliminación de etiquetas de la cuenta de usuario, así como las funciones de etiquetado y visualización de etiquetas asociadas a recursos.
- 4) **Aplicación móvil – Modelo de datos:** Implementación de las funciones que permiten hacer persistentes todos los datos de la cuenta de un usuario. Esto implica la creación de las operaciones del *Web service* que permiten conectar con la base de datos y añadir, modificar y eliminar aquellos registros asociados a los *workspaces*, recursos y etiquetas manejados a partir de la aplicación móvil.
- 5) **Aplicación web – Multi-tool:** Implementación de un método que permita al usuario elegir el tipo de *workspace* a crear: *group sketching* o diagrama de afinidad. En función del tipo de *workspace* creado, la interfaz del mismo es diferente.
- 6) **Aplicación web – Diagrama de afinidad:** Implementación de las funciones básicas de un diagrama de afinidad. Incluye la creación de ítems de información, la agrupación de los mismos y la disolución de los grupos de ítems.
- 7) **Aplicación web – Recursos multimedia:** Implementación del módulo de recursos en la aplicación web. Supone la inclusión de un nuevo espacio en el que se presenten los recursos añadidos por el usuario a través de la aplicación móvil, así como la función de añadir y visualizar recursos en un ítem de información.
- 8) **Aplicación web – Modelo de datos:** Implementación de las funciones que permiten hacer persistentes los datos de un diagrama de afinidad. Esto implica la creación de las operaciones del *Web service* que permiten conectar con la base de datos y añadir, modificar y eliminar aquellos registros asociados a los ítems de información y los grupos de ítems creados en un diagrama de afinidad.

Anexo III. Especificación de requisitos

Tras haber definido en la sección *Definición de requisitos* los requisitos que ha de cumplir el sistema a desarrollar, es necesario detallar la información de estos requisitos, con el fin de evitar posibles ambigüedades o incoherencias. Para ello, se hace uso de la plantilla propuesta en la *Tabla 10*.

En primer lugar, se detallan los requisitos funcionales que deben cumplir la aplicación web y la aplicación móvil y, a continuación, se presentan los requisitos no funcionales que deben cumplir la aplicación web y la aplicación móvil.

Requisitos funcionales

Los requisitos funcionales que debe cumplir la aplicación web son los siguientes:

Identificador	RF-W-1		
Prioridad	Alta	Necesidad	Esencial
Descripción	Las salas de la antigua aplicación pasarán a llamarse <i>workspace</i> .		
Prueba asociada	Comprobar que todas las ocurrencias de “sala” han sido sustituidas por “workspace”.		

Identificador	RF-W-2		
Prioridad	Alta	Necesidad	Esencial
Descripción	El usuario podrá crear dos tipos de <i>workspace</i> : para la realización de <i>group sketching</i> , y para la creación de diagramas de afinidad. Los <i>workspace</i> para la realización de <i>group sketching</i> se corresponderán con las antiguas salas.		
Prueba asociada	Comprobar que antes de crear un nuevo <i>workspace</i> se puede elegir el tipo de <i>workspace</i> a crear.		

Identificador	RF-W-3		
Prioridad	Alta	Necesidad	Deseable
Descripción	Las opciones de creación de un <i>workspace</i> para la creación de diagramas de afinidad serán las mismas que para la creación de las antiguas salas de ColDes.		
Prueba asociada	Comparar las opciones de creación de las antiguas salas y las opciones de creación de los <i>workspace</i> de diagrama de afinidad, y comprobar que son iguales.		

Identificador	RF-W-4		
Prioridad	Alta	Necesidad	Deseable
Descripción	Las opciones de colaboración de los <i>workspace</i> para la creación de diagramas de afinidad serán las mismas que en las antiguas salas de ColDes.		
Prueba asociada	Comprobar que dentro de un <i>workspace</i> el usuario puede escoger entre colaboración “uno a uno” y colaboración “todos a la vez”.		

Identificador	RF-W-5		
Prioridad	Alta	Necesidad	Esencial
Descripción	En un <i>workspace</i> para la creación de diagramas de afinidad, el usuario podrá crear ítems de información.		
Prueba asociada	Comprobar que el usuario puede crear ítems de información.		

Identificador	RF-W-6		
Prioridad	Alta	Necesidad	Deseable
Descripción	Todos los ítems de información creados por el usuario contendrán información textual.		
Prueba asociada	Comprobar que al crear un ítem de información se le pide al usuario que introduzca la información textual asociada al mismo.		

Identificador	RF-W-7		
Prioridad	Alta	Necesidad	Deseable
Descripción	Cualquier usuario que pertenezca a un <i>workspace</i> para la creación de diagramas de afinidad podrá modificar el diagrama asociado al mismo.		
Prueba asociada	Crear un <i>workspace</i> en el que participen varios usuarios. Comprobar que todos los participantes pueden modificar el diagrama de afinidad asociado al <i>workspace</i> .		

Identificador	RF-W-8		
Prioridad	Media	Necesidad	Deseable
Descripción	El usuario podrá modificar la posición de los ítems de información dentro del espacio de trabajo del <i>workspace</i> .		
Prueba asociada	Comprobar que el posible desplazar un ítem de información dentro del área de trabajo del <i>workspace</i> .		

Identificador	RF-W-9		
Prioridad	Baja	Necesidad	Opcional
Descripción	El usuario podrá redimensionar los ítems de información dentro del espacio de trabajo del <i>workspace</i> .		
Prueba asociada	Comprobar que es posible cambiar las dimensiones de los ítems de información de un diagrama de afinidad.		

Identificador	RF-W-10		
Prioridad	Baja	Necesidad	Deseable
Descripción	El usuario podrá elegir el color de los ítems de información.		
Prueba asociada	Comprobar que cualquier usuario que pertenezca a un <i>workspace</i> puede elegir el color asociado a cualquier ítem de información.		

Identificador	RF-W-11		
Prioridad	Media	Necesidad	Deseable
Descripción	El usuario podrá eliminar los ítems creados anteriormente.		
Prueba asociada	Comprobar que cualquier usuario que pertenezca a un <i>workspace</i> puede eliminar cualquiera de los ítems de un diagrama de afinidad.		

Identificador	RF-W-12		
Prioridad	Alta	Necesidad	Esencial
Descripción	El usuario podrá agrupar los ítems de información con contenido similar en grupos de ítems.		
Prueba asociada	Probar que dentro de un <i>workspace</i> de diagrama de afinidad existe una función que permita al usuario agrupar visualmente varios ítems de información.		

Identificador	RF-W-13		
Prioridad	Media	Necesidad	Deseable
Descripción	El usuario podrá disolver un grupo de ítems de información.		
Prueba asociada	Comprobar que dentro de un <i>workspace</i> de diagrama de afinidad existe una función que permita al usuario desagrupar ítems de información que se encuentren agrupados.		

Identificador	RF-W-14		
Prioridad	Alta	Necesidad	Esencial
Descripción	El usuario podrá dar un título a un grupo de ítems. Este título será identificativo para el grupo.		
Prueba asociada	Comprobar que dentro de un <i>workspace</i> de diagrama de afinidad existe una función que permita al usuario añadir un título a un grupo de ítems de información.		

Identificador	RF-W-15		
Prioridad	Media	Necesidad	Deseable
Descripción	El usuario podrá guardar el diagrama de afinidad asociado a un <i>workspace</i> .		
Prueba asociada	Comprobar que existe una función que permite al usuario guardar el estado actual de un diagrama de afinidad.		

Identificador	RF-W-16		
Prioridad	Media	Necesidad	Deseable
Descripción	El usuario podrá cargar el diagrama de afinidad asociado a un <i>workspace</i> .		
Prueba asociada	Probar que el usuario puede cargar el diagrama de afinidad guardado anteriormente para un <i>workspace</i> .		

Identificador	RF-W-17		
Prioridad	Alta	Necesidad	Esencial
Descripción	El usuario podrá acceder a un espacio donde se encuentren los recursos externos.		
Prueba asociada	Comprobar que desde la página de inicio de un usuario se puede acceder a un espacio donde se listen los recursos externos asociados a su cuenta.		

Identificador	RF-W-18		
Prioridad	Media	Necesidad	Opcional
Descripción	El usuario podrá asociar un recurso de sus recursos externos a un ítem de información.		
Prueba asociada	Comprobar que existe una función dentro de un ítem de información que permita al usuario escoger de su lista de recursos externos aquel		

	que desee incluir al ítem.
--	----------------------------

Identificador	RF-W-19		
Prioridad	Media	Necesidad	Opcional
Descripción	Los recursos añadidos a una nota adhesiva podrán ser de tipo audio, vídeo, imagen o texto.		
Prueba asociada	Comprobar que la lista de recursos externos de un usuario puede contener recursos de tipo audio, vídeo, imagen o texto.		

Identificador	RF-W-20		
Prioridad	Media	Necesidad	Opcional
Descripción	El usuario podrá eliminar el recurso asociado a una nota adhesiva.		
Prueba asociada	Comprobar que existe una función dentro de un ítem de información que permita al usuario eliminar un recurso asociado al mismo.		

Los requisitos funcionales que debe cumplir la aplicación móvil son los siguientes:

Identificador	RF-M-1		
Prioridad	Alta	Necesidad	Esencial
Descripción	Para hacer uso de la aplicación móvil, el usuario debe tener una cuenta en la aplicación web de ColDes.		
Prueba asociada	Probar que el usuario no puede acceder a la aplicación móvil si no tiene cuenta en la aplicación web.		

Identificador	RF-M-2		
Prioridad	Alta	Necesidad	Deseable
Descripción	El acceso a la aplicación se realizará mediante un componente de autenticación que pedirá al usuario el nombre y la contraseña.		
Prueba asociada	Comprobar que antes de acceder a su cuenta, se le pide al usuario su nombre y contraseña.		

Identificador	RF-M-3		
Prioridad	Baja	Necesidad	Opcional
Descripción	El usuario podrá cambiar sus datos de usuario desde la aplicación móvil.		
Prueba asociada	Comprobar que existe una función que permite al usuario modificar los		

	datos de usuario asociado a su cuenta.
--	--

Identificador	RF-M-4		
Prioridad	Alta	Necesidad	Esencial
Descripción	El usuario podrá añadir recursos de audio, vídeo, imagen y/o texto a su cuenta.		
Prueba asociada	Comprobar que existe una zona dedicada a añadir recursos a la cuenta de usuario.		

Identificador	RF-M-5		
Prioridad	Media	Necesidad	Deseable
Descripción	Al añadir un recurso, se pedirá al usuario el nombre del recurso y un comentario para el mismo.		
Prueba asociada	Comprobar que antes de añadir un recurso el usuario debe haber introducido el nombre del recurso y el comentario asociado al mismo.		

Identificador	RF-M-6		
Prioridad	Media	Necesidad	Opcional
Descripción	El usuario podrá obtener los recursos de imagen y vídeo desde la galería del dispositivo móvil o desde la cámara.		
Prueba asociada	Comprobar que se le da la opción al usuario de subir archivos de imagen y vídeo tanto de la galería del móvil como directamente de la cámara.		

Identificador	RF-M-7		
Prioridad	Alta	Necesidad	Esencial
Descripción	El usuario podrá ver la lista de los recursos añadidos a su cuenta.		
Prueba asociada	Comprobar que al acceder a su cuenta, el usuario puede acceder a una zona en la que se listen los recursos que ha añadido a su cuenta.		

Identificador	RF-M-8		
Prioridad	Media	Necesidad	Esencial
Descripción	El usuario podrá visualizar el contenido de cualquier recurso de la lista de recursos.		
Prueba asociada	Probar que en la zona en la que se listan los recursos, el usuario puede pulsar sobre un recurso para visualizar su contenido.		

Identificador	RF-M-9		
Prioridad	Baja	Necesidad	Deseable
Descripción	El usuario podrá eliminar cualquier recurso de su lista de recursos.		
Prueba asociada	Comprobar que dentro de la lista de recursos es posible eliminar cualquiera de ellos.		

Identificador	RF-M-10		
Prioridad	Alta	Necesidad	Esencial
Descripción	El usuario podrá crear etiquetas para asociarlas a los recursos.		
Prueba asociada	Probar que dentro de su cuenta, el usuario puede acceder a una zona en la que se permita la creación de etiquetas.		

Identificador	RF-M-11		
Prioridad	Alta	Necesidad	Deseable
Descripción	Al crear una etiqueta, se le pedirá al usuario el nombre de la etiqueta y el color asociado.		
Prueba asociada	Comprobar que el usuario no puede crear una etiqueta sin haber introducido un nombre y un color para la misma.		

Identificador	RF-M-12		
Prioridad	Alta	Necesidad	Esencial
Descripción	El usuario podrá ver la lista de etiquetas añadidas a su cuenta.		
Prueba asociada	Comprobar que dentro de su cuenta, el usuario puede acceder a una zona en la que aparezca la lista de etiquetas añadidas a su cuenta.		

Identificador	RF-M-13		
Prioridad	Alta	Necesidad	Esencial
Descripción	El usuario podrá asociar etiquetas de su lista a etiquetas a los recursos de su cuenta.		
Prueba asociada	Comprobar que al acceder a un recurso el usuario puede asociar cualquiera de las etiquetas de su lista al recurso.		

Identificador	RF-M-14		
Prioridad	Alta	Necesidad	Esencial

Descripción	El usuario podrá ver la lista de etiquetas asociadas a un recurso concreto.
Prueba asociada	Comprobar que al acceder a un recurso el usuario puede ver la lista de etiquetas asociadas al mismo.

Identificador	RF-M-15		
Prioridad	Alta	Necesidad	Esencial
Descripción	El usuario podrá ver la lista de recursos etiquetados por una etiqueta dada.		
Prueba asociada	Comprobar que al acceder a la información de una etiqueta de la lista de etiquetas aparecen todos los recursos etiquetados por esa etiqueta.		

Identificador	RF-M-16		
Prioridad	Media	Necesidad	Opcional
Descripción	El usuario podrá eliminar cualquier etiqueta creada anteriormente.		
Prueba asociada	Comprobar que el usuario puede eliminar cualquier etiqueta de su lista de etiquetas.		

Requisitos no funcionales

Los requisitos no funcionales que debe cumplir la aplicación web son los siguientes:

Identificador	RNF-W-1		
Prioridad	Alta	Necesidad	Esencial
Descripción	Un <i>workspace</i> para la creación de diagramas de afinidad permitirá crear un único diagrama de afinidad.		
Prueba asociada	Comprobar que no es posible crear más de un diagrama de afinidad en un mismo <i>workspace</i> .		

Identificador	RNF-W-2		
Prioridad	Media	Necesidad	Opcional
Descripción	El aspecto de los ítems del diagrama de afinidad será similar al de una nota adhesiva.		
Prueba asociada	Comprobar que el aspecto de los ítems de un diagrama de afinidad se corresponde al de notas adhesivas.		

Identificador	RNF-W-3		
Prioridad	Media	Necesidad	Opcional
Descripción	El color de los ítems se escogerá a partir de un selector de colores incluido en el propio ítem.		
Prueba asociada	Comprobar que el usuario puede cambiar el color de un ítem de información a partir de un selector de colores.		

Identificador	RNF-W-4		
Prioridad	Alta	Necesidad	Esencial
Descripción	Se permitirá añadir un único recurso a un ítem de información.		
Prueba asociada	Comprobar que no es posible añadir más de un recurso a un ítem de información.		

Los requisitos no funcionales que debe cumplir la aplicación móvil son los siguientes:

Identificador	RNF-M-1		
Prioridad	Baja	Necesidad	Deseable
Descripción	La aplicación deberá soportar los idiomas inglés y español.		
Prueba asociada	Comprobar que el idioma de la aplicación se adapta al idioma del dispositivo móvil para los idiomas inglés y español.		

Identificador	RNF-M-2		
Prioridad	Alta	Necesidad	Esencial
Descripción	La aplicación estará optimizada para su ejecución en dispositivos Android con una versión del sistema operativo 4.0 o superior.		
Prueba asociada	Comprobar que la aplicación funciona correctamente en dispositivos con una versión Android 4.0 o superior.		

Identificador	RNF-M-3		
Prioridad	Baja	Necesidad	Deseable
Descripción	La aplicación debe estar optimizada para las densidades de pantalla <i>mdpi</i> , <i>hdpi</i> y <i>xdpi</i> .		
Prueba asociada	Comprobar que la interfaz de la aplicación se adapta correctamente a dispositivos con distinta densidad de pantalla.		

Identificador	RNF-M-4		
Prioridad	Alta	Necesidad	Esencial
Descripción	La conexión con la base de datos de realizará por medio de un <i>Web service</i> implementado en lenguaje Java.		
Prueba asociada	Comprobar que la conexión con la base de datos se realiza por medios de un <i>Web service</i> en lenguaje Java.		

Anexo IV. Prototipo

En el presente capítulo se presenta el prototipo realizado para la aplicación móvil, ya que el prototipo de la aplicación web ya se presentó por completo en el apartado *Diseño detallado*.

Aplicación móvil

A continuación se muestran las imágenes de la segunda iteración de prototipado de la aplicación móvil, agrupadas por los módulos de información que representan.

- **Módulo de autenticación:** Se incluyen los prototipos de las interfaces que permiten al usuario introducir sus datos de usuario y acceder al sistema o, en caso contrario, los mensajes de error pertinentes.
- **Módulo de recursos:** Se incluyen los prototipos de las interfaces que permiten al usuario obtener una lista de los recursos asociados a su cuenta, así como añadir nuevos recursos.
- **Vista de recurso:** Se incluyen los prototipos de las interfaces que permiten al usuario ver el contenido concreto de un recurso, así como las etiquetas asociadas al mismo. Igualmente, se muestra la interfaz que permite al usuario añadir nuevas etiquetas.
- **Módulo de usuario:** Se incluyen los prototipos de las interfaces que permiten al usuario modificar sus datos de usuario.
- **Módulo de *workspaces*:** Se incluyen los prototipos de las interfaces que permiten al usuario visualizar la lista de *workspaces* a los que pertenece, así como modificar los datos de cualquiera de ellos, y crear nuevos.
- **Vista de *workspace*:** Se incluyen los prototipos de las interfaces que permiten al usuario ver los recursos asociados al *workspace* seleccionado, así como asociar nuevos recursos al mismo.



Figura 37: Prototipo de la interfaz del módulo de autenticación de la aplicación móvil. Segunda iteración.

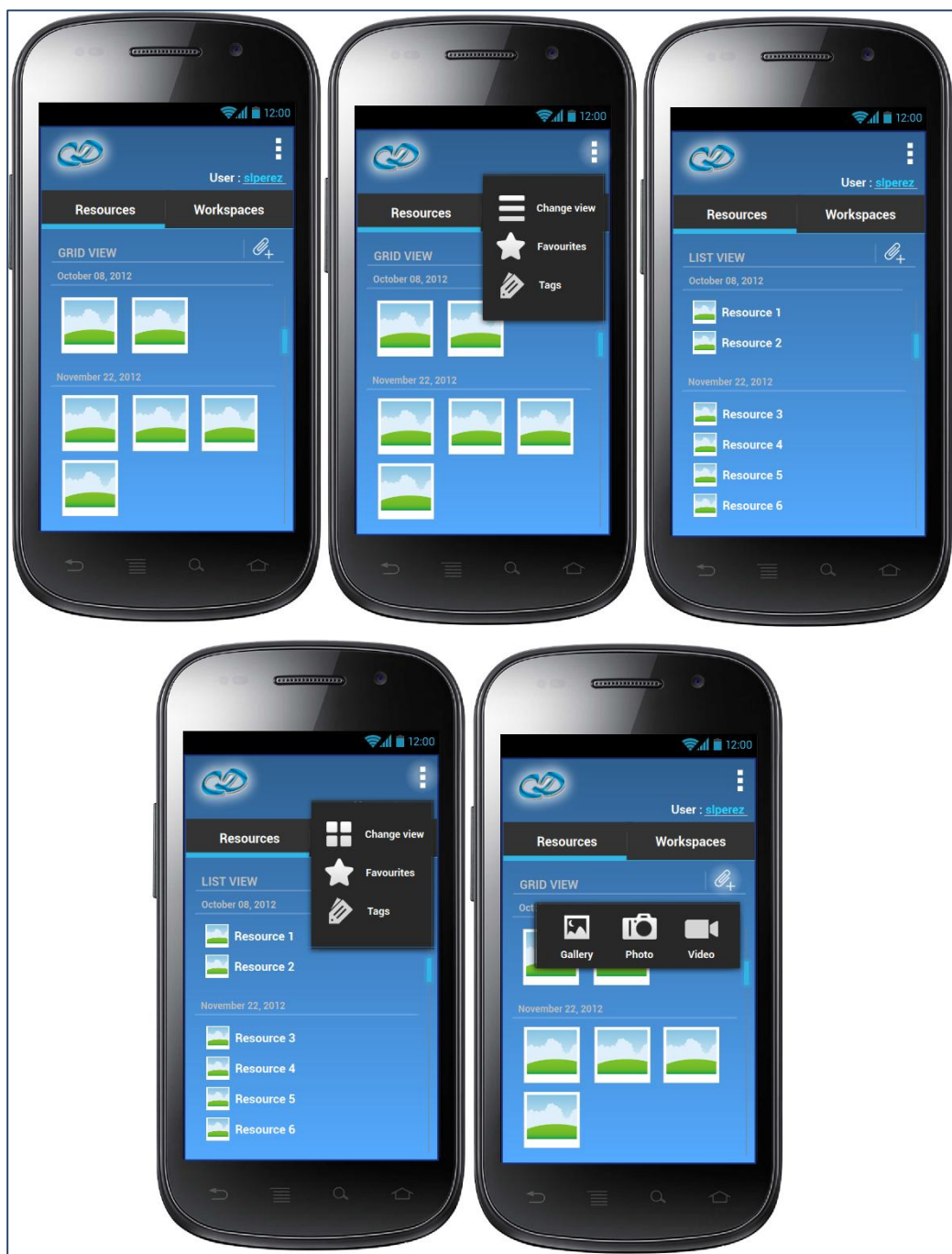


Figura 38: Prototipo de la interfaz del módulo de recursos de la aplicación móvil. Segunda iteración.



Figura 39: Prototipo de la interfaz de la vista de recursos de la aplicación móvil. Segunda iteración.



Figura 40: Prototipo de la interfaz del módulo de etiquetas de la aplicación móvil. Segunda iteración.



Figura 41: Prototipo de la interfaz del módulo de usuario de la aplicación móvil. Segunda iteración.

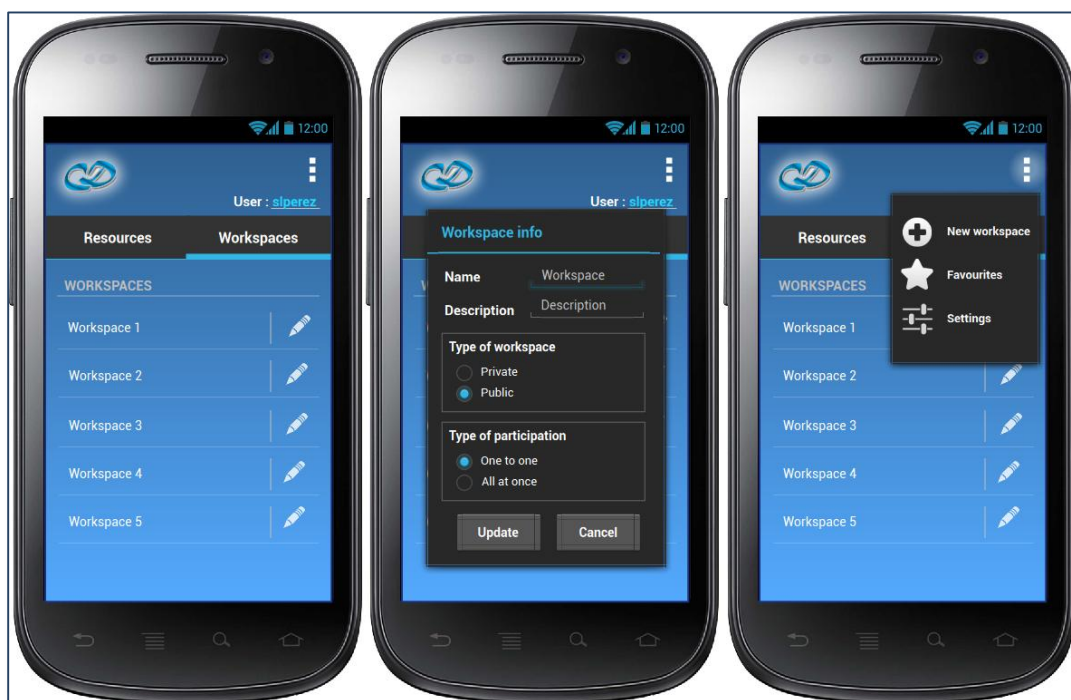


Figura 42: Prototipo de la interfaz del módulo de *workspaces* de la aplicación móvil. Segunda iteración.



Figura 43: Prototipo de la interfaz de la vista de *workspace* de la aplicación móvil. Segunda iteración.